

КВАЛІФІКАЦІЙНА РОБОТА

Метод інтеграції технологій машинного навчання у програмну систему
управління бізнесом шляхом точкової автоматизації бізнес-процесів

Рівень вищої освіти Другий (магістерський)

Галузь знань 12 «Інформаційні технології»

Спеціальність 121 «Інженерія програмного забезпечення»

Освітня програма Освітньо-професійна програма «Інженерія програмного
забезпечення»

Шифр КвРПЗ.2301113.01.07.ПЗ

Виконав студент 2 курсу, група ІПЗм-23-1


Підпис

Василь КОГУТ
Ім'я, ПРІЗВИЩЕ

Керівник д-р фіз.-мат. наук, професор
Науковий ступінь, звання


Підпис

Леонід БЕДРАТЮК
Ім'я, ПРІЗВИЩЕ

Нормоконтролер к. пед. наук, доцент


Підпис

Наталія ПРАВОРСЬКА
Ім'я, ПРІЗВИЩЕ

До захисту допускаю:

Завідувач кафедри інженерії
програмного забезпечення


Підпис

Леонід БЕДРАТЮК
Ім'я, ПРІЗВИЩЕ

2 грудня 2024 р.

Хмельницький 2024

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет Інформаційних технологій
Кафедра Інженерії програмного забезпечення
Рівень вищої освіти Другий (магістерський)
Галузь знань 12 «Інформаційні технології»
Спеціальність 121 «Інженерія програмного забезпечення»
Освітня програма Освітньо-професійна програма «Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ

Завідувач кафедри ПЗ

Л. П. Бедратюк

02.09.2024 р.

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

Когут Василь Сергійович

Прізвище, ім'я, по батькові (додрук)

1. Тема роботи Метод інтеграції технологій машинного навчання у програмні системи
бізнесом шляхом точкової автоматизації бізнес-процесів

Керівник роботи Бедратюк Леонід Петрович, д-р фіз.-мат. наук, професор

Прізвище, ім'я, по батькові, науковий ступінь, вчене звання

Затверджена наказом ректора університету від 26.08.2024 р. № 60

2. Строк подання студентом роботи на кафедру 02.12.2024 р.

3. Вихідні дані до роботи Матеріали науково-дослідної практики

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити)

1. Теоретичний виклад досліджуваної проблеми

2. Метод інтеграції машинного навчання для автоматизації бізнес-процесів

3. Архітектура програмної реалізації

4. Програмна реалізація

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслень)

Презентаційні матеріали (слайди)

	консультанта	завдання видали	завдання прийняв
Антиплагіат	доцент К. П. Н. Ю. В.		
Нормоконтроль	доцент, к. ед. наук Трощевська Н. В.		

7. Дата видачі завдання « 02 » вересня 2024 р.

КАЛЕНДАРНИЙ ПЛАН

Назва етапів (розділів) кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1 Вивчення предметної області; формулювання мети та задач дослідження; визначення об'єкта та предмета дослідження;	02.09-10.09.2024	
2 Робота над розділом 1 кваліфікаційної роботи – вивчення літературних та Інтернет-джерел; аналіз відомих моделей, методів та засобів за темою роботи;	11.09-25.09.2024	
3 Робота над розділом 2 кваліфікаційної роботи – розробка моделей, методів та алгоритмів вирішення задачі; висновки до розділу	26.09-10.10.2024	
4 Робота над науковими статтями	11.10-30.10.2024	
5 Робота над розділом 3 кваліфікаційної роботи – розробка інформаційної технології вирішення задачі	11.10-26.10.2024	
6 Робота над розділом 4 кваліфікаційної роботи – програмна реалізація спроектованих рішень, результати експериментів та їх аналіз;	27.10-17.11.2024	
7 Попередній захист кваліфікаційної роботи	Листопад (згідно графіка)	
8 Узгодження постановки задачі, отриманих результатів та висновків; оформлення пояснювальної записки та графічних матеріалів згідно вимог чинних стандартів	18.11-30.11.2024	
9 Перевірка роботи на наявність плагіату; нормоконтроль; брошурування пояснювальної записки; підготовка супровідних документів	02.12-04.12.2024	
10 Підготовка до захисту кваліфікаційної роботи	з 02.12.2024 р.	

Студент Василь Шаєт
Ім'я, ПРІЗВИЩЕ

Керівник роботи Леонід Федорук
Ім'я, ПРІЗВИЩЕ

РЕФЕРАТ

Тема кваліфікаційної роботи: «Метод інтеграції технологій машинного навчання у програмну систему управління бізнесом шляхом точкової автоматизації бізнес-процесів.»

Автор роботи: Когут Василь.

Керівник роботи: Бедратюк Леонід Петрович.

Пояснювальна записка: 114 с., 8 рис., 4 табл., 3 дод., 25 джерел.

АВТОМАТИЗАЦІЯ БІЗНЕС-ПРОЦЕСІВ, ІНТЕГРАЦІЯ МАШИННОГО НАВЧАННЯ, НЕЙРОННІ МЕРЕЖІ, МІКРОСЕРВІСНА АРХІТЕКТУРА, ПОПЕРЕДНЯ ОБРОБКА ДАНИХ, ЦІЛЬОВА АВТОМАТИЗАЦІЯ, ПРИЙНЯТТЯ РІШЕНЬ НА ОСНОВІ МАШИННОГО НАВЧАННЯ, ВДОСКОНАЛЕННЯ БІЗНЕС-ПРОГРАМ.

Мета роботи – розроблення методу інтеграції машинного навчання у програмні системи управління бізнесом для забезпечення адаптивної точкової автоматизації бізнес-процесів і оптимізації ключових управлінських етапів.

Предмет – методи інтеграції технологій машинного навчання для точкової автоматизації бізнес-процесів, спрямовані на підвищення ефективності управлінських рішень.

Об'єкт – процеси автоматизації бізнес-процесів у програмних системах управління.

На основі мети, предмету, об'єкту кваліфікаційної роботи магістра можна виділити наступні завдання дослідження:

- Провести аналіз сучасних методів автоматизації бізнес-процесів із використанням машинного навчання.
- Визначити ключові бізнес-процеси, які найбільше потребують точкової автоматизації.
- Розробити алгоритми автоматизації обраних бізнес-процесів із використанням моделей машинного навчання.
- Інтегрувати розроблений алгоритм у програмну систему управління бізнесом.

– Оцінити ефективність запропонованого методу на основі експериментальних даних.

Наукова новизна отриманих результатів полягає в тому що вперше розроблено метод інтеграції машинного навчання у програмні системи управління бізнесом, який дозволяє здійснювати точкову автоматизацію бізнес-процесів для підвищення адаптивності до змін ринкових умов. Також удосконалено підходи до адаптації моделей машинного навчання у бізнес-середовищах, що забезпечують оптимізацію процесів, які потребують інтенсивної обробки даних.

У даній кваліфікаційній роботі магістра здійснено розробку методу інтеграції машинного навчання у програмні системи управління бізнесом для автоматизації процесів розподілу платежів та оптимізації фінансових потоків. Результатом цієї роботи є створення програмної системи на основі нейронної мережі, яка дозволяє підвищити точність і швидкість обробки даних, мінімізувати людський фактор та автоматизувати процеси прийняття рішень. Результати дослідження можуть бути застосовані для удосконалення існуючих бізнес-процесів у фінансовій сфері, сприяючи підвищенню ефективності управління. Доцільність та ефективність розробленого підходу підтверджено результатами емпіричних досліджень, описаних у роботі.

В якості теоретичних методів дослідження взято: аналіз, синтез, моделювання, порівняння.

Емпіричними методами дослідження виступають: опис та тестування.

28.11.2029

В. С. Сидор

ABSTRACT

Topic of the Qualification Work: “Method of Integrating Machine Learning Technologies into Business Management Software Systems through Targeted Automation of Business Processes.”

Author: Vasyl Kohut.

Supervisor: Leonid Petrovych Bedratiuk.

Explanatory Note: 114 pages, 8 figures, 4 tables, 3 appendices, 25 references.

Keywords: BUSINESS PROCESS AUTOMATION, MACHINE LEARNING INTEGRATION, NEURAL NETWORKS, MICROSERVICES ARCHITECTURE, DATA PREPROCESSING, TARGETED AUTOMATION, ML-DRIVEN DECISION-MAKING, BUSINESS SOFTWARE ENHANCEMENT.

Objective: To develop a method for integrating machine learning into business management software systems to ensure adaptive targeted automation of business processes and optimize key management stages.

Subject: Methods of integrating machine learning technologies for targeted automation of business processes aimed at improving managerial decision-making efficiency.

Object: Automation processes of business processes in business management software systems.

Based on the objective, subject, and object of the master’s qualification work, the following research tasks were identified:

- Analyze modern methods of business process automation using machine learning.
 - Identify key business processes most in need of targeted automation.
 - Develop algorithms for automating selected business processes using machine learning models.
 - Integrate the developed algorithm into a business management software system.
 - Assess the effectiveness of the proposed method based on experimental data.
 - Scientific Novelty:
-
-

The scientific novelty of the obtained results lies in the development of a method for integrating machine learning into business management software systems, allowing for targeted automation of business processes to enhance adaptability to market changes. Additionally, approaches to adapting machine learning models in business environments have been improved to optimize processes requiring intensive data processing.

This master's qualification work developed a method for integrating machine learning into business management software systems to automate payment distribution processes and optimize financial flows. The result of this work is the creation of a software system based on a neural network, enabling improved accuracy and speed of data processing, minimizing human factors, and automating decision-making processes. The research results can be applied to enhance existing business processes in the financial sector, contributing to improved management efficiency. The feasibility and effectiveness of the developed approach have been confirmed by empirical research described in the work.

Theoretical Research Methods Used: Analysis, synthesis, modeling, comparison.

Empirical Research Methods Used: Description and testing.

28.11.2024

B. Kaidun

ЗМІСТ

ВСТУП	8
1 ТЕОРЕТИЧНИЙ ВИКЛАД ДОСЛІДЖУВАНОЇ ПРОБЛЕМИ	11
1.1 Аналіз предметної області	11
1.2 Аналіз існуючих рішень	12
1.3 Огляд методів вирішення проблеми	18
1.4 Постановка задачі	21
1.5 Висновки	22
2 МЕТОД ІНТЕГРАЦІЇ МАШИННОГО НАВЧАННЯ ДЛЯ АВТОМАТИЗАЦІЇ БІЗНЕС-ПРОЦЕСІВ	24
2.1 Теоретична модель автоматизації бізнес-процесів.....	24
2.2 Математичні та об'єктно-орієнтовані моделі бізнес-процесів.....	29
2.3 Методи навчання та адаптації моделей машинного навчання	34
2.4 Висновки.....	39
3 АРХІТЕКТУРА ПРОГРАМНОЇ РЕАЛІЗАЦІЇ	42
3.1 Формування та аналіз вимог програмної реалізації системи	42
3.2 Вибір та обґрунтування моделі машинного навчання	44
3.3 Розробка алгоритмів для автоматизованого розподілу платежів.....	52
3.4 Архітектурний та компонентний дизайн системи	58
3.5 Технічні вимоги до програмних та апаратних засобів	62
3.6 Висновок	63
4 ПРОГРАМНА РЕАЛІЗАЦІЯ	66
4.1 Програмна реалізація.....	66
4.1.1 Мікросервіс обробки платіжних транзакцій	66
4.1.2 Мікросервіс класифікації інвойсів	67
4.1.3 Мікросервіс для управління базою	70
4.1.4 Мікросервіс для тренування та дотреновування нейронної мережі.	71
4.2 Тестування та верифікація системи	74
4.3 Аналіз ефективності запропонованого підходу	82
4.4 Інтеграція в існуючі бізнес-процеси	85
4.5 Висновки.....	87
ВИСНОВКИ	89
ДОДАТОК А	84
ДОДАТОК Б	94
ДОДАТОК В	96

ВСТУП

Автоматизація бізнес-процесів є важливим напрямом у розвитку сучасного управління, що надає можливість компаніям суттєво підвищити ефективність, зменшити витрати та оптимізувати використання ресурсів. Сучасні дослідження свідчать, що більшість компаній уже застосовують автоматизацію у своїх процесах, але традиційні методи не завжди враховують необхідність гнучкості в умовах швидких змін ринку. Зокрема, McKinsey [1]. повідомляє, що близько 60% завдань у бізнесі можуть бути автоматизовані. Впровадження технологій штучного інтелекту (ШІ) та машинного навчання (ML) дозволяє підвищити точність, швидкість і гнучкість бізнес-процесів, оптимізуючи їх адаптацію до нових умов.

Актуальність роботи полягає у створенні методів точкової автоматизації на основі технологій машинного навчання, що дозволяють адаптуватися до ринкових змін і забезпечувати ефективне управління ресурсами. Сучасні системи автоматизації управління бізнесом часто побудовані на основі статичних алгоритмів, що ускладнює їхнє застосування в умовах динамічних ринкових змін. Використання машинного навчання дозволяє створювати більш гнучкі системи, здатні адаптуватися до нових вимог та змін у реальному часі. Водночас, існують проблеми, пов'язані з ідентифікацією точок, де ML може бути найбільш ефективним. Таким чином, необхідно розробити метод інтеграції ML, що дозволить використовувати точкову автоматизацію, зосереджуючись лише на ключових процесах, які потребують оптимізації.

Виконана дипломна робота має взаємозв'язок з такими науковими напрямами та роботами, як інноваційні методи автоматизації бізнес-процесів, які досліджуються кафедрою інженерії програмного забезпечення, що зосереджується на застосуванні ML для адаптивного управління даними та підвищення ефективності у великих бізнес-структурах. Проект орієнтований на використання технологій ML для забезпечення адаптивності і гнучкості в управлінських системах, що відповідає програмам досліджень кафедри, спрямованим на інтеграцію сучасних технологій у бізнес-середовище.

Метою дипломної роботи є розробка методу інтеграції машинного навчання у програмні системи управління бізнесом, що дозволяє реалізувати точкову автоматизацію бізнес-процесів для оптимізації ключових етапів управління.

Поставлена мета досягається розв'язанням таких основних задач:

1 Аналіз існуючих методів автоматизації бізнес-процесів з акцентом на використання ML.

2 Ідентифікація критичних точок, де впровадження ML може забезпечити максимальну ефективність.

3 Розробка алгоритмів машинного навчання для конкретних бізнес-процесів.

4 Інтеграція моделей ML у програмні системи управління бізнесом.

5 Оцінка ефективності розробленого методу через практичне тестування.

Об'єктом дослідження є процес автоматизації бізнес-процесів у системах управління, який характеризується необхідністю швидкої адаптації до змін і потребує оптимізації.

Предметом дослідження є властивості методів точкової автоматизації бізнес-процесів із використанням технологій машинного навчання, які забезпечують вирішення задачі оптимізації бізнесу і підвищення ефективності управлінських рішень.

Для досягнення мети використані наступні методи: аналіз наукової літератури для вивчення поточного стану дослідження в галузі ML і бізнес-автоматизації; синтез і моделювання для розробки алгоритмів ML; емпіричне тестування, яке дозволило перевірити ефективність розробленого методу у реальних бізнес-сценаріях.

Наукова новизна отриманих результатів: вперше одержано метод інтеграції машинного навчання у програмні системи управління бізнесом з акцентом на точкову автоматизацію бізнес-процесів, що дозволяє досягти адаптивності до змін ринкових умов; удосконалено підхід до визначення точок автоматизації за допомогою методів ML, що підвищує ефективність управлінських рішень у критичних етапах; отримано подальший розвиток методів адаптації моделей ML у

бізнес-середовищах для оптимізації процесів, що потребують інтенсивної обробки даних.

Практична цінність отриманих результатів полягає у можливості використання розробленого методу для впровадження ML у системи управління бізнесом, забезпечуючи автоматизацію критичних процесів і підвищення ефективності управлінських рішень. Запропонований підхід дозволяє компаніям адаптувати свої операційні системи до сучасних умов ринку, знижуючи витрати та оптимізуючи ресурси.

Отримані результати можуть бути застосовані при реалізації систем управління у сфері логістики, управління запасами, фінансах, прогнозуванні попиту тощо.

За темою кваліфікаційної роботи опубліковані тези «Метод підтримки якості програмного коду на основі автоматичного тестування» на конференції «Актуальні проблеми комп'ютерних наук АПКН-2024».

1 ТЕОРЕТИЧНИЙ ВИКЛАД ДОСЛІДЖУВАНОЇ ПРОБЛЕМИ

1.1 Аналіз предметної області

Автоматизація бізнес-процесів протягом останніх десятиліть стала важливою частиною управління бізнесом, дозволяючи зменшити витрати та підвищити продуктивність. Однак, для того щоб системи автоматизації працювали ефективно, необхідно правильно визначити проблеми, які вони вирішують, і завдання, які повинні бути виконані. Основні методологічні підходи у цій галузі розглядають бізнес як сукупність процесів, які можуть бути оптимізовані через автоматизацію, зокрема за рахунок використання інтелектуальних моделей на базі машинного навчання (ML). У контексті інтеграції машинного навчання (ML) у системи автоматизації існує кілька важливих аспектів, що вимагають детального дослідження.

Автоматизація бізнесу охоплює широкий спектр завдань, які включають:

- Управління ресурсами: Контроль за використанням матеріальних, фінансових та людських ресурсів є критичним завданням для будь-якого бізнесу. Автоматизація допомагає зменшити витрати на управління та підвищити ефективність їх використання.

- Документообіг та обмін інформацією: Зокрема для великих підприємств, ефективність документообігу є важливою умовою швидкості прийняття рішень і загальної продуктивності.

- Аналітика та прийняття рішень: Автоматизовані системи здатні швидко обробляти величезні обсяги даних і надавати аналітичну інформацію, що може бути використана для прийняття управлінських рішень. Однак людський фактор, особливо при нестандартних рішеннях, все ще залишається важливим.

- Одним із ключових викликів автоматизації бізнесу є необхідність глибокого вивчення процесів, які будуть автоматизовані. Проблеми виникають, коли опис процесу не є достатньо точним або містить певні приховані деталі, які не завжди доступні для розробників або технічних спеціалістів. Для ефективної автоматизації

необхідно провести докладний аналіз всіх кроків процесу і визначити місця, де можливе використання машинного навчання для оптимізації.

Наявні проблеми включають:

- Неоднорідність даних: У процесі прийняття рішень системи машинного навчання можуть зіткнутися з проблемою неоднорідних вхідних даних. Це може вплинути на точність моделі і відповідно на якість автоматизованого процесу.

- Різноманітність бізнес-процесів: У кожному бізнесі є своя специфіка, тому системи автоматизації повинні враховувати різні форми та структури процесів.

- Людський фактор: Незважаючи на автоматизацію, деякі рішення залишаються залежними від людини, особливо коли необхідно враховувати нестандартні ситуації або приймати творчі рішення.

1.2 Аналіз існуючих рішень

Інтеграція технологій машинного навчання у бізнес-процеси є одним із сучасних напрямків розвитку автоматизації управлінських і виробничих систем. Ринок рішень для автоматизації бізнес-процесів значно розширився за останні роки, пропонуючи як комплексні системи для великих корпорацій, так і вузькоспеціалізовані інструменти для малого та середнього бізнесу. Можна виділити такі наявні рішення:

- Системи управління бізнес-процесами (BPM - Business Process Management). Конкретне визначення для управління бізнес-процесами (BPM) від Європейської асоціації управління бізнес-процесами (EABPM) описує BPM як систематичний підхід для захоплення, формування, виконання, документування, вимірювання, моніторингу та керування автоматичними й неавтоматичними процесами для досягнення узгоджених та стійких цілей компанії [1]. Мета BPM полягає в підвищенні корпоративної продуктивності шляхом оптимізації та управління бізнес-процесами компанії[2]. Управління бізнес-процесами можна описати як оптимізацію і управління корпоративними процесами на рівні окремих систем, таких як ERP, CRM і SCM, а також у межах цілісної інтегрованої мережі.

ВРМ-система також призначена для координації та покрокового виконання бізнес-процесів [2]. Завдяки моніторингу, виявленню та оцінці проблем у процесах, ВРМ-система дозволяє визначити, де виникла проблема, і допомагає бізнесу оптимізувати цей процес [2]. Сучасні системи управління бізнес-процесами (ВРМ) є комплексними рішеннями, що дозволяють компаніям автоматизувати, керувати та аналізувати їхні процеси. Основна перевага таких систем полягає в тому, що вони надають компаніям можливість інтегрувати всі процеси в одну систему, забезпечуючи прозорість і контроль на всіх етапах бізнесу. IBM Business Automation Workflow: інтегрована платформа для автоматизації робочих процесів і управління бізнесом, яка дозволяє легко створювати рішення для управління процесами на основі правил, а також підтримує інтеграцію з машинним навчанням. Arripan: хмарна платформа для створення рішень ВРМ, що пропонує інтеграцію з технологіями штучного інтелекту та машинного навчання. Arripan дозволяє компаніям автоматизувати процеси на основі даних та аналізувати їх у реальному часі.

- ERP-системи (Enterprise Resource Planning) ERP-системи забезпечують управління основними бізнес-процесами підприємств, такими як фінанси, виробництво, логістика, HR тощо. В останні роки в такі системи активно інтегруються модулі на основі машинного навчання для покращення прогнозування попиту, оптимізації логістики та управління запасами. SAP S/4HANA: сучасна ERP-платформа, яка включає вбудовані функції машинного навчання для аналізу даних і прийняття рішень у реальному часі. Вона допомагає компаніям знижувати витрати та підвищувати ефективність, автоматизуючи складні бізнес-процеси. Oracle ERP Cloud: рішення, яке використовує аналітику та машинне навчання для прогнозування фінансових ризиків і автоматизації планування ресурсів.

- CRM-системи (Customer Relationship Management) CRM-системи використовуються для управління взаємодією з клієнтами, зберігання даних про продажі та аналіз поведінки клієнтів. Використання машинного навчання в CRM-системах. Прогнозування поведінки клієнтів: Машинне навчання допомагає CRM-

системам аналізувати великі обсяги історичних даних і прогнозувати дії клієнтів. Це включає такі аспекти, як оцінка ймовірності покупки, прогнозування клієнтської лояльності або ймовірності відтоку клієнтів. В основі цих прогнозів лежать алгоритми, які використовують змінні, такі як поведінка клієнтів у минулому, соціально-демографічні характеристики та інші ключові показники (KPIs). Персоналізація маркетингових пропозицій: Інтелектуальні алгоритми аналізують індивідуальні потреби клієнтів і автоматично генерують персоналізовані пропозиції. Наприклад, такі платформи, як Salesforce Einstein, інтегрують штучний інтелект для створення індивідуальних рекомендацій та автоматизації взаємодії з клієнтами. Ці рекомендації можуть бути застосовані до рекламних кампаній, що підвищує ефективність продажів і задоволення клієнтів. Оцінка та відбір потенційних клієнтів (lead scoring): CRM-системи з машинним навчанням, такі як Zoho CRM, використовують автоматизовані інструменти для ранжування потенційних клієнтів на основі ймовірності конверсії. Це дозволяє ефективніше витратити ресурси на клієнтів, які мають найбільші шанси стати постійними покупцями. Аналіз настроїв і тональності: Деякі CRM-системи інтегрують алгоритми обробки природної мови (NLP), що дозволяють аналізувати тональність текстів із соціальних мереж, повідомлень клієнтів або відгуків. Це дає змогу компаніям виявити настрої клієнтів, зрозуміти їхню реакцію на продукти і послуги та своєчасно реагувати на негативні відгуки або запити клієнтів. Salesforce Einstein: рішення, яке використовує штучний інтелект для автоматизації взаємодії з клієнтами, прогнозування поведінки клієнтів та оптимізації маркетингових кампаній. Zoho CRM: це платформа з інтегрованими інструментами машинного навчання для аналізу поведінки клієнтів, оцінки потенційних клієнтів та автоматизації продажів. HubSpot CRM: Ще одна популярна платформа, яка використовує машинне навчання для автоматизації різних процесів, таких як персоналізація електронних листів, сегментація клієнтів і прогнозування продажів. HubSpot аналізує поведінку клієнтів на сайті, у соціальних мережах та інших каналах, що дозволяє налаштувати маркетингові кампанії відповідно до інтересів кожного користувача. Microsoft Dynamics 365: Ця платформа інтегрує потужні

аналітичні інструменти для обробки великих обсягів даних і дозволяє користувачам отримувати аналітику в реальному часі. Вбудований інструмент AI Customer Insights застосовує машинне навчання для виявлення шаблонів у поведінці клієнтів і автоматизації персоналізованих пропозицій.

- Спеціалізовані рішення для автоматизації бізнес-процесів У світі також існує багато спеціалізованих рішень, які орієнтовані на автоматизацію конкретних бізнес-процесів з використанням машинного навчання. UiPath: провайдер рішень для роботизованої автоматизації процесів (RPA), що дозволяє автоматизувати повторювані ручні задачі. UiPath активно інтегрує машинне навчання для обробки складних процесів, що включають аналіз документів або взаємодію з некерованими даними. Automation Anywhere: рішення, яке дозволяє автоматизувати бізнес-процеси на основі штучного інтелекту та машинного навчання, пропонуючи можливості аналізу великих обсягів даних та побудови аналітичних моделей для прогнозування. Blue Prism Ще одна популярна платформа, яка дозволяє компаніям інтегрувати RPA з машинним навчанням. Вона пропонує функції для створення аналітичних звітів, управління ризиками та обробки складних задач з автоматизованим ухваленням рішень. Kofax Орієнтується на автоматизацію завдань, пов'язаних із обробкою зображень, оптичним розпізнаванням символів (OCR) та інтеграцією з ERP-системами. Завдяки штучному інтелекту Kofax оптимізує процеси документообігу та скорочує витрати на обробку даних. Ці спеціалізовані рішення надають бізнесу гнучкість у виконанні складних задач і дозволяють знизити трудовитрати, оптимізуючи автоматизацію для конкретних процесів. Такі інструменти все частіше використовуються для інтеграції з ERP, CRM та іншими корпоративними системами, розширюючи їх функціонал і дозволяючи компаніям швидше реагувати на динамічні зміни в бізнесі.

- Системи машинного навчання та AI як додатки для автоматизації бізнеспроцесів Сучасні системи машинного навчання також використовуються для автоматизації процесів, що потребують аналітичних здібностей або інтелектуальних прогнозів. Google Cloud AI та Azure AI: ці платформи пропонують моделі машинного навчання, які можуть інтегруватися в існуючі системи

автоматизації для виконання специфічних завдань, таких як обробка зображень, прогнозування на основі даних або автоматизація процесів на основі мовних команд. Проаналізувавши існуючі рішення, можна зробити висновок, що більшість сучасних систем автоматизації бізнесу вже активно інтегрують технології машинного навчання. Проте, кожне рішення має свої особливості, і вибір платформи чи підходу залежить від специфіки бізнесу, а також від поставлених завдань. Технології машинного навчання дозволяють не лише автоматизувати існуючі процеси, але й значно підвищувати їхню ефективність шляхом аналізу великих даних, прогнозування та адаптації до змінних умов.

Системи управління корпоративними даними (EDM) традиційно спираються на ручні процеси та алгоритми, засновані на правилах, для обробки даних. Незважаючи на те, що ці системи були основоположними в організації та зберіганні інформації, вони мають властиві обмеження, які стримують їхню масштабованість, можливість ухвалення рішень в режимі реального часу та загальну ефективність. Традиційні методи управління корпоративними даними (EDM) давно є основою стратегій обробки даних в організаціях. Однак ці методи дедалі частіше стикаються з викликами, пов'язаними з їхньою нездатністю адаптуватися до динамічних умов сучасних підприємств. Таблиця 1.1 нижче підсумовує основні недоліки як традиційних EDM-систем, так і систем на основі штучного інтелекту.

Таблиця 1.1 - Недоліки традиційних систем EDM та систем на основі ШІ

Категорія	Традиційні EDM системи	Системи на основі ШІ
Якість даних	Ручне введення даних призводить до непослідовностей та помилок.	Фрагментарна автоматизація може не вирішити всіх проблем якості даних.
Масштабованість	Обмежена масштабованість зі збільшенням обсягів даних.	Масштабованість залежить від інфраструктури та складності моделі.

Робота в режимі реального часу	Пакетна обробка обмежує прийняття рішень в режимі реального часу.	Можливості в режимі реального часу можуть варіюватися, вимагаючи додаткової інтеграції.
Інтеграція	Розрізнені сховища даних ускладнюють доступ до міжфункціональних інсайтів.	Проблеми інтеграції з різними джерелами даних.
Складність	Спирається на алгоритми, засновані на правилах; бракує адаптивного навчання.	Потребує експертизи в галузі науки про дані та розгортання моделей ШІ.
Вартість	Низькі початкові інвестиції, але вищі операційні витрати.	Вищі початкові інвестиції через інфраструктуру та навчання.

Наявні системи, такі як CRM і ERP, забезпечують управління клієнтською базою або фінансовими даними, але вони часто недостатньо гнучкі для врахування специфіки окремих бізнесів. Використання машинного навчання для точкової автоматизації — інноваційний підхід, який дозволить враховувати попередні знання системи і автоматизувати прийняття рішень, що, на сьогоднішній день, у більшості систем автоматизації не є стандартною функцією.

1.3 Огляд методів вирішення проблеми

Завдання автоматизації бізнес-процесів за допомогою машинного навчання сьогодні розв'язують завдяки різноманітним методам, які дозволяють ефективно обробляти великі обсяги інформації, передбачати фінансові показники та оптимізувати розподіл ресурсів. У сучасній практиці ці методи використовуються для аналітичної обробки та класифікації транзакцій, що надає можливість виявляти важливі фінансові патерни та вчасно реагувати на зміни в бізнес-середовищі. Одним із найпоширеніших підходів є використання алгоритмів машинного навчання для прогнозування та класифікації даних. Зокрема, такі моделі, як лінійна та нелінійна регресія, дозволяють передбачати поведінку фінансових показників на основі аналізу минулих даних, що є основою для подальшої оптимізації бюджетів і витрат. Крім того, популярними є дерева рішень і методи ансамблю, які, поєднуючи різні моделі, знижують ризик помилок і підвищують надійність результатів, особливо у разі складних взаємозалежностей між даними. Для побудови гнучкої і масштабованої системи автоматизації доречно застосовувати мікросервісну архітектуру, яка дає змогу розділити систему на самостійні частини, кожна з яких відповідає за конкретні завдання. Це включає модулі обробки транзакцій, які забезпечують прийняття, сортування та класифікацію фінансових операцій, а також модулі прогнозування, які обчислюють найбільш оптимальні напрямки для подальших дій на основі результатів машинного навчання. Це дозволить досягти не тільки оптимізації стандартних завдань, але й підвищення ефективності в прийнятті рішень на основі аналізу даних. Одним із ключових аспектів інтеграції машинного навчання у системи автоматизації бізнесу є модульність розроблених рішень. Це означає, що автоматизація бізнес-процесів та впровадження машинного навчання можуть бути впроваджені поетапно, як окремі компоненти програмного забезпечення. Кожен модуль може бути відповідальним за окремий процес, а технології машинного навчання можна застосовувати у точках прийняття рішень або на етапах аналізу даних. Такий підхід дозволяє:

- Легко інтегрувати нові процеси в існуючі системи без необхідності масштабної реконструкції.

- Застосовувати машинне навчання тільки в критичних точках бізнеспроцесів, де це приносить найбільшу користь. Поступово вдосконалювати та адаптувати моделі машинного навчання на основі нових даних, не порушуючи роботи інших компонентів системи.

Застосування машинного навчання на етапі прийняття рішень Основною перевагою технологій машинного навчання є їх здатність виявляти закономірності в даних і робити прогнози на їх основі. У бізнес-процесах це може бути особливо корисним для автоматизації таких завдань, як:

- Прогнозування попиту на продукцію або послуги. - Виявлення потенційних ризиків у ланцюжку постачання.

- Аналіз клієнтської поведінки та персоналізація пропозицій. Методологічно цей підхід включає:

- Збір та аналіз даних: На початковому етапі необхідно зібрати великі масиви даних, які будуть використовуватися для навчання моделей машинного навчання.

- Навчання моделей: На основі зібраних даних створюються та навчання моделі машинного навчання, що зможуть приймати рішення або давати прогнози в рамках конкретного бізнес-процесу.

- Інтеграція результатів: Отримані результати застосовуються до автоматизованої системи, де модель машинного навчання може виконувати або підтримувати прийняття рішень у певних точках бізнес-процесів.

Використання ітеративного підходу для донавчання моделей Застосування машинного навчання в системах автоматизації бізнесу повинно мати ітеративний характер. Це означає, що моделі повинні бути не тільки створені на основі наявних даних, але й постійно донавчатися на нових даних, що надходять у процесі експлуатації системи. Ітеративний підхід передбачає:

- Моніторинг та оцінка ефективності: Регулярне тестування моделей для визначення їхньої точності та продуктивності.

- Донавчання моделей: Моделі повинні бути періодично донавчені на основі нових даних або після значних змін у бізнес-процесах, щоб зберігати їхню актуальність та точність.

- Адаптація до нових умов: Моделі машинного навчання повинні бути адаптивними до змін у зовнішньому та внутрішньому середовищі бізнесу. Це може включати врахування нових трендів на ринку або зміни в структурі клієнтів.

Впровадження та інтеграція з існуючими системами управління Одним із ключових завдань є інтеграція систем машинного навчання з наявними програмними рішеннями, які використовуються в управлінні бізнесом. Це може бути реалізовано за допомогою API або інших інтерфейсів, що дозволяють безперешкодний обмін даними між системами автоматизації та модулями машинного навчання. Такий підхід дозволяє:

- Зберегти спадкоємність між старими системами та новими компонентами.
 - Оптимізувати взаємодію між різними частинами бізнес-процесів.
- Підвищити ефективність автоматизації шляхом інтеграції інтелектуальних рішень на основі машинного навчання.

Участь фахівців з різних сфер для точкової автоматизації та впровадження рішень на основі машинного навчання також передбачає спільну роботу різних фахівців: програмістів, бізнес-аналітиків, інженерів даних, фахівців із машинного навчання та менеджерів проєктів. Кожен з них грає важливу роль на різних етапах автоматизації, від аналізу процесів до розробки і впровадження програмних рішень. Спільна робота дозволяє:

- Максимально точно визначити задачі, які слід автоматизувати.
- Використати фахову експертизу для побудови якісних моделей машинного навчання. Забезпечити ефективне впровадження нових технологій у бізнес-процеси, що знижує ризики та підвищує продуктивність.

1.4 Постановка задачі

У рамках цього дослідження постає завдання розробки методу, який базується на застосуванні машинного навчання для автоматизації конкретних бізнес-процесів та оптимізації управлінських рішень. Основні задачі, які потребують вирішення, включають:

- Визначення бізнес-процесів для автоматизації: Аналіз існуючих бізнес-процесів компанії та виявлення ключових етапів, які піддаються точковій автоматизації з використанням технологій машинного навчання. Це включає процеси, що найбільше потребують оптимізації через їх високу ресурсозатратність або значні часові витрати.

- Збір та аналіз історичних даних: Збір та обробка даних, що відображають минулі рішення у подібних бізнес-ситуаціях. Це включає аналіз даних про час виконання окремих процесів, витрати на їх реалізацію та ризики, з якими стикався бізнес. Ці дані допоможуть створити моделі для передбачення результатів виконання завдань у майбутньому.

- Розробка алгоритмів для прийняття рішень: Створення та адаптація алгоритмів машинного навчання, які дозволять прогнозувати результати бізнеспроцесів, оцінювати їхню ефективність та оптимізувати витрати. Алгоритми повинні враховувати специфіку кожного процесу та бути гнучкими для подальшого навчання на нових даних.

- Донавчання моделей машинного навчання: Впровадження ітеративного підходу до навчання та донавчання моделей на основі нових даних і змін у бізнес-процесах. Це забезпечить актуальність моделі та її адаптивність до нових умов роботи.

- Інтеграція машинного навчання у системи автоматизації бізнесу: Розробка рішень для безшовної інтеграції моделей машинного навчання у вже існуючі системи управління бізнесом. Це дозволить підвищити якість управлінських рішень на основі аналітичних прогнозів, що надаватимуть моделі. Завдання полягає у створенні комплексного підходу, що не тільки підвищить ефективність окремих

автоматизованих бізнес-процесів, але й надасть можливість використовувати машинне навчання для поліпшення якості прийнятих рішень. Це дозволить зменшити витрати на бізнес-операції, знизити рівень ризиків і підвищити загальну продуктивність організації.

1.5 Висновки

Розділ підсумовує існуючі рішення у сфері автоматизації бізнес-процесів з використанням машинного навчання (ML) і штучного інтелекту (AI). Огляд показав, що сьогодні широко застосовуються CRM-системи (наприклад, Salesforce Einstein, Zoho CRM) для управління взаємодією з клієнтами, а також спеціалізовані платформи для автоматизації бізнес-процесів, такі як UiPath та Automation Anywhere, що використовують AI для обробки великих обсягів даних та забезпечення ефективного керування процесами.

Автоматизація бізнес-процесів є ключовим елементом управління бізнесом, що підвищує продуктивність і знижує витрати. Для ефективності таких систем потрібна чітка постановка завдань та оптимізація процесів із застосуванням машинного навчання. Основні проблеми автоматизації включають неоднорідність даних, специфіку процесів і залежність від людського фактору, що підкреслює потребу у глибокому аналізі та точковій інтеграції ML для більшої автономності систем.

Існуючі рішення, незважаючи на широкий функціонал і можливості, мають свої обмеження, зокрема у складності інтеграції, масштабованості та необхідності в людському контролі у нестандартних ситуаціях. Це вказує на необхідність подальших досліджень для розробки більш адаптивних і всеосяжних систем, здатних обробляти неоднорідні дані, забезпечувати кращу інтеграцію різних процесів та знижувати залежність від людського втручання.

Актуальність проблеми обумовлена прагненням бізнесів підвищити ефективність і точність управління процесами за допомогою сучасних технологій, що підвищує доцільність проведення подальших досліджень у цій галузі. Об'єктом

дослідження є системи автоматизації бізнес-процесів, а предметом — методи інтеграції машинного навчання у такі системи. Мета дослідження полягає у розробці оптимального методу інтеграції технологій ML для підвищення ефективності автоматизації бізнес-процесів, а завдання — дослідити поточні рішення, визначити їхні обмеження та запропонувати удосконалений підхід для вирішення виявлених проблем.

Автоматизація бізнес-процесів з використанням машинного навчання реалізується через різноманітні методи, що сприяють ефективній обробці даних, прогнозуванню фінансових показників і оптимізації ресурсів. Ці методи включають аналітичну обробку та класифікацію транзакцій для виявлення фінансових патернів і швидкої реакції на зміни в бізнес-середовищі. Найпоширенішими є алгоритми для прогнозування і класифікації даних, зокрема регресія та дерева рішень, які допомагають знизити ризики помилок і підвищити надійність результатів. Для побудови масштабованих систем рекомендується мікросервісна архітектура, яка дозволяє модульно інтегрувати машинне навчання, що забезпечує гнучкість та адаптивність у бізнес-процесах. Ітеративний підхід до навчання моделей та інтеграція з існуючими системами управління підвищують ефективність автоматизації, залучаючи фахівців з різних галузей для точкового впровадження технологій.

Дослідження фокусується на розробці методу автоматизації бізнес-процесів за допомогою машинного навчання, що включає кілька ключових завдань. Спочатку необхідно визначити бізнес-процеси для автоматизації, особливо ті, що мають високі витрати. Потім слід зібрати та проаналізувати історичні дані для прогнозування результатів. Далі важливо розробити гнучкі алгоритми машинного навчання та впровадити ітеративний підхід до їх донавчання. Завершується процес інтеграцією машинного навчання в існуючі системи управління, що підвищить ефективність рішень, зменшить витрати та ризики, і покращить загальну продуктивність організації.

2 МЕТОД ІНТЕГРАЦІЇ МАШИННОГО НАВЧАННЯ ДЛЯ АВТОМАТИЗАЦІЇ БІЗНЕС-ПРОЦЕСІВ

2.1 Теоретична модель автоматизації бізнес-процесів

Теоретична модель автоматизації бізнес-процесів за допомогою технологій машинного навчання включає всебічний підхід до аналізу, оптимізації та автоматизації основних етапів, які впливають на ефективність управлінських рішень. Модель передбачає побудову концептуальної структури, що охоплює всі аспекти функціонування бізнесу: фінансові, операційні та стратегічні процеси, з метою досягнення оптимізації продуктивності. На початковому етапі проводиться аналіз існуючих бізнес-процесів і визначення найбільш ресурсозатратних етапів, які доцільно автоматизувати через застосування машинного навчання.

В основу моделі закладено інтеграцію алгоритмів машинного навчання, які виконуватимуть завдання класифікації, прогнозування та виявлення аномалій у різних операційних процесах. Наприклад, класифікаційні алгоритми можуть бути застосовані для автоматизованого аналізу транзакцій або фінансових операцій, що дозволить більш оперативно виявляти потенційні аномалії. Моделі прогнозування, навчені на історичних даних, допомагають визначити тренди та прогнозувати показники, що є необхідними для стратегічного планування. При цьому, важливо, щоб теоретична модель підтримувала можливість регулярного оновлення та донавчання алгоритмів, що дозволить адаптувати їх до змін ринкових умов та нових даних.

Основа моделі також складає модульний підхід, що дозволяє структурувати систему на незалежні компоненти. Кожен модуль обслуговує окремий бізнес-процес або групу процесів, що спрощує обслуговування та вдосконалення системи. Наприклад, модуль для обробки транзакцій може функціонувати окремо від модуля прогнозування, що дозволяє модифікувати або вдосконалювати їх незалежно один від одного. Така архітектура полегшує інтеграцію нових алгоритмів або моделей машинного навчання в існуючу систему без потреби в складних змінах структури.

Необхідним елементом моделі є управління даними, оскільки основою для навчання алгоритмів є якісні, структуровані та консистентні дані. У процесі автоматизації бізнес-процесів важливо враховувати специфіку великих обсягів даних, що надходять з різних джерел, і забезпечити єдиний підхід до їх обробки та зберігання. Розробка єдиного стандарту управління даними дозволить стабілізувати роботу модулів і забезпечити їхню ефективність в прийнятті рішень.

Також модель включає розробку показників для оцінки ефективності автоматизації. Такими показниками можуть бути точність прогнозів, швидкість обробки транзакцій, рівень виявлення аномалій, економія ресурсів тощо. Моніторинг індикаторів ефективності дозволяє оцінювати роботу моделі та своєчасно приймати рішення про донавачання або модифікацію алгоритмів, що є необхідним для забезпечення актуальності та якості автоматизації.

Основні елементи автоматизованих бізнес-процесів у контексті розподілу коштів можна розглядати через кілька ключових компонентів, кожен із яких має свою специфіку та потребує відповідного алгоритмічного підходу для ефективного функціонування. Першим елементом є прийняття рішень, що лежить в основі процесу оптимального розподілу коштів, особливо коли цей розподіл базується на попередніх даних. Алгоритми машинного навчання, такі як регресійні моделі та нейронні мережі, можуть бути використані для прогнозування майбутніх значень та вибору найкращого сценарію розподілу, враховуючи історичні закономірності. Наприклад, застосування нейронної мережі дозволяє аналізувати минулі транзакції та розробити оптимальні схеми витрат, підвищуючи точність рішень щодо подальшого фінансового планування. Другим важливим елементом є обробка транзакцій, яка забезпечує швидке та точне виконання фінансових операцій і може бути автоматизована за допомогою алгоритмів класифікації та контролю якості. Дерева рішень або алгоритми підтримки векторів дозволяють класифікувати транзакції за пріоритетністю, обсягом або іншими параметрами, що забезпечує гнучкість у налаштуванні процесу обробки фінансових операцій залежно від потреб бізнесу. Третій елемент — класифікація та прогнозування результатів розподілу, що дозволяє передбачати найбільш вигідні фінансові рішення в різних

умовах. Алгоритми класифікації, такі як k-ближчих сусідів або ансамблеві методи, забезпечують розподіл даних за категоріями, що є критично важливим для формування найбільш ефективної стратегії витрат і мінімізації ризиків. Прогнозування на основі регресійних моделей допомагає виявити фінансові тренди та скоригувати розподіл коштів у майбутньому відповідно до актуальних ринкових умов. Четвертий аспект — обґрунтування вибору алгоритмів для кожного з компонентів. Так, нейронні мережі добре підходять для глибокого аналізу складних даних і часто використовуються в процесах, де важливо виявити приховані залежності між транзакціями. Деревя рішень ефективні для прийняття рішень з багаторівневою класифікацією, що забезпечує швидкість та надійність у прийнятті рішень. Використання регресійних моделей, у свою чергу, виправдано там, де важливо передбачати числові результати, наприклад, суму витрат на певні напрямки діяльності. Загалом, ефективна автоматизація процесу розподілу коштів потребує системного підходу до вибору алгоритмів, щоб кожен етап виконувався з максимальною точністю, що дозволить досягти високої ефективності та економічної результативності фінансового управління.

Розробка теоретичної моделі автоматизації охоплює весь цикл інтеграції машинного навчання в бізнес-процеси. Початковий етап — це визначення основних бізнес-операцій, які можуть бути автоматизовані, а також оцінка їхньої ресурсозатратності. На цьому етапі важливо розуміти, як кожен процес функціонує в поточній системі, щоб точно визначити, які операції потребують автоматизації для зниження навантаження на людський ресурс та підвищення продуктивності. Наприклад, розподіл коштів у фінансових операціях часто є складним процесом, який залежить від багатьох факторів, таких як історичні дані, прогнозовані доходи та видатки, що потребує складних обчислень та може бути значно оптимізованим за допомогою алгоритмів машинного навчання.

Наступним етапом є інтеграція алгоритмів для виконання задач класифікації, прогнозування та виявлення аномалій. У даному випадку класифікаційні алгоритми можуть допомогти не тільки в аналізі транзакцій, але й у сортуванні різних фінансових операцій за пріоритетністю або категоріями, що спрощує

фінансовий контроль та сприяє зменшенню ризиків. Алгоритми прогнозування, навчені на попередніх даних, дозволяють створювати достовірні прогнози для оцінки можливих фінансових сценаріїв. Наприклад, можна передбачати зміни в доходах чи витратах компанії, що особливо корисно при розподілі бюджетів.

Ще одним важливим аспектом є забезпечення модульності структури системи. Модульний підхід дозволяє розділити систему на окремі функціональні блоки, кожен з яких відповідає за певний етап бізнес-процесу, наприклад, за обробку транзакцій, прогнозування фінансових показників, або ж аналіз ризиків. Така модульна архітектура надає гнучкості системі: якщо потрібно оновити чи замінити певний алгоритм, це можна зробити локально, не впливаючи на загальну функціональність. З точки зору практичності, цей підхід забезпечує масштабованість системи та можливість інтеграції нових алгоритмів без необхідності внесення масштабних змін в існуючу інфраструктуру.

Не менш важливим є управління даними — зокрема, організація потоку даних, забезпечення їхньої якості та актуальності, адже саме на цих даних навчаються алгоритми. Важливо розробити єдиний підхід до зберігання, обробки та оновлення даних, що забезпечить стабільну роботу моделей. У контексті фінансових процесів управління даними включає збір інформації з різних джерел, забезпечення її консистентності та, при необхідності, нормалізацію, щоб уникнути надмірного дублювання та забезпечити ефективну роботу алгоритмів машинного навчання.

Окрім того, важливим елементом моделі є створення системи показників ефективності автоматизації. Для оцінки продуктивності та доцільності автоматизації можуть бути використані різні індикатори, такі як точність прогнозів, швидкість обробки транзакцій, рівень економії ресурсів, а також рівень виявлення аномалій. Ці показники дозволяють регулярно контролювати якість роботи алгоритмів та вчасно реагувати на зміни, що є особливо важливим для забезпечення надійності моделі в динамічному бізнес-середовищі.

Таким чином, теоретична модель автоматизації бізнес-процесів за допомогою технологій машинного навчання включає розробку комплексного

підходу до аналізу, оптимізації та автоматизації ключових етапів управлінських процесів. Вона поєднує в собі різноманітні алгоритмічні підходи для реалізації автоматизації прийняття рішень, обробки транзакцій, класифікації та прогнозування фінансових операцій, забезпечуючи інтеграцію алгоритмів у модульну структуру з постійним моніторингом та оцінкою ефективності.

Одним із ключових аспектів теоретичної моделі автоматизації бізнес-процесів є інтеграція технологій машинного навчання таким чином, щоб вони не тільки підвищували ефективність існуючих процесів, але й створювали нові можливості для бізнесу. Наприклад, застосування глибоких нейронних мереж може дозволити підприємствам виявляти приховані закономірності у фінансових даних, які недоступні через традиційний аналіз, що відкриває перспективи для нових напрямків стратегічного планування.

Модель повинна включати елемент управління ризиками, особливо у фінансових операціях. За допомогою алгоритмів кластеризації можна створювати профілі ризиків для різних типів транзакцій або клієнтів, а алгоритми виявлення аномалій можуть оперативно сигналізувати про підозрілі операції. Це дозволить компаніям підвищити рівень безпеки та зменшити фінансові втрати через шахрайство або людські помилки.

Критично важливою є здатність моделі до адаптації в умовах динамічних ринків. Для цього в систему інтегруються механізми автоматичного оновлення алгоритмів на основі нових даних. Регулярний процес донавчання дозволяє моделі залишатися актуальною та ефективною навіть за умов швидких змін, таких як поява нових регуляторних вимог чи зміна споживчих уподобань.

Технології штучного інтелекту відкривають нові можливості у стратегічному плануванні. Наприклад, алгоритми прогнозування можуть визначати довгострокові тенденції розвитку ринку, що дозволяє компаніям проактивно реагувати на потенційні зміни та коригувати свої плани. Такий підхід підвищує конкурентоспроможність бізнесу та забезпечує стійкість до зовнішніх викликів.

Інтеграція машинного навчання в бізнес-процеси також сприяє покращенню аналітичних інструментів. Автоматизований аналіз великих масивів даних може

надати глибшу аналітику щодо продуктивності кожного підрозділу компанії, допомагаючи виявляти слабкі місця та шукати шляхи їх покращення. Це дозволяє керівникам приймати більш обґрунтовані рішення на основі даних.

Для ефективного функціонування моделі доцільно розглянути використання хмарних обчислень. Це забезпечить масштабованість системи, швидкий доступ до обчислювальних ресурсів і можливість зберігання великих обсягів даних. Хмарна архітектура також полегшує інтеграцію з іншими бізнес-системами та платформами, що підвищує загальну ефективність автоматизації.

Автоматизовані системи повинні бути орієнтованими на кінцевого користувача. Інтуїтивний інтерфейс, прозорі алгоритми прийняття рішень і зрозумілий механізм роботи сприяють кращому прийняттю системи співробітниками та підвищують їхню залученість до процесу. Це також знижує бар'єри для інтеграції нових технологій у корпоративну культуру.

У майбутньому модель може бути розширена через інтеграцію додаткових технологій, таких як Інтернет речей (IoT) для автоматичного збору даних або блокчейн для підвищення прозорості фінансових операцій. Такі розширення створюють можливості для інновацій та підтримують довгострокову еволюцію бізнес-процесів.

2.2 Математичні та об'єктно-орієнтовані моделі бізнес-процесів

Математичні моделі є важливою складовою для опису бізнес-процесів, оскільки вони дозволяють формалізувати взаємозв'язки між різними компонентами процесів і надають можливість використовувати точні методи для їхньої оптимізації. У контексті автоматизації фінансових операцій, математичні моделі можуть включати використання лінійної або багатофакторної регресії для прогнозування показників на основі попередніх даних. Такі моделі ефективні для вирішення задач, пов'язаних із розподілом коштів, аналізом витрат, а також оптимізацією рішень щодо інвестування. Математичні методи дозволяють

створювати моделі, які враховують змінні фактори, як-то ринкові умови, попит на послуги чи товари, і допомагають у прийнятті оптимальних фінансових рішень.

Об'єктно-орієнтовані моделі, у свою чергу, дозволяють представити бізнес-процеси як сукупність об'єктів та їхніх взаємодій. Кожен об'єкт може мати певні атрибути (наприклад, клієнт, рахунок, транзакція), а також методи взаємодії з іншими об'єктами. У рамках моделювання автоматизації розподілу фінансів об'єктно-орієнтований підхід дозволяє структурувати бізнес-процеси як взаємодію окремих модулів або компонентів. Наприклад, модуль, що відповідає за обробку платежів, може взаємодіяти з модулем, що управляє прогнозуванням розподілу коштів, при цьому кожен з модулів виконує свою конкретну задачу.

Центральним елементом таких моделей є об'єктні відношення між учасниками процесу, наприклад, між платниками і отримувачами, що дозволяє створювати точні моделі розподілу фінансових потоків. Використання діаграм класів та діаграм послідовностей у системному проектуванні дозволяє чітко визначити сценарії взаємодії об'єктів і, відповідно, алгоритми автоматизації кожного етапу процесу. Об'єктно-орієнтовані моделі також дозволяють враховувати спадковість і поліморфізм, що полегшує повторне використання модулів або компонентів при модифікації системи.

У моделюванні бізнес-процесів особливе значення мають стохастичні моделі, які використовуються для врахування випадкових змінних або невизначеностей, що часто зустрічаються у фінансових операціях. Такі моделі дозволяють автоматизованим системам краще адаптуватися до мінливих умов, зокрема до непередбачуваних змін у ринковій кон'юнктурі або до коливань попиту. До основних стохастичних підходів належать моделі Марковських процесів, які можуть застосовуватися для прогнозування динаміки ринку або змін у поведінці клієнтів.

Описані математичні та об'єктно-орієнтовані підходи формують основу для створення гнучкої автоматизованої системи управління бізнес-процесами, що дозволяє оптимізувати прийняття рішень і забезпечує високу точність у розподілі ресурсів. Математичні моделі допомагають у формалізації процесів, тоді як

об'єктно-орієнтовані моделі дозволяють структурувати та спростити складні взаємодії між учасниками бізнес-процесу, що особливо важливо в контексті великомасштабних систем. Для автоматизації бізнес-процесів у сфері управління фінансами, особливо коли йдеться про оптимальний розподіл коштів на основі попередніх даних, застосування математичних та об'єктно-орієнтованих моделей є критично важливим, оскільки вони підвищують точність, ефективність і адаптивність всієї системи. Математичні моделі у цьому контексті дозволяють побудувати чітку формалізацію процесів, що підлягають оптимізації, за допомогою обчислювальних методів. Такі моделі допомагають виявляти закономірності у великих масивах фінансових даних, що особливо важливо для автоматизованого аналізу та прогнозування фінансових результатів. Регресійні моделі, наприклад, дозволяють передбачати можливі майбутні надходження та витрати на основі історичних даних, що стає ключовим у прийнятті рішень щодо розподілу ресурсів. Завдяки регресії можна визначити залежності між різними фінансовими показниками, як-от витрати і доходи, а також оцінити їхній взаємозв'язок з іншими економічними змінними, такими як ринкові умови або сезонність. Простий підхід через лінійну регресію допомагає оцінити тренди у витратах та доходах за певними категоріями, що полегшує бюджетування та планування витрат. У випадках, коли залежності є більш складними, застосовуються нелінійні моделі або багатофакторна регресія, що підвищує точність передбачень і дозволяє побудувати повноцінну модель фінансових операцій.

Ще одним важливим інструментом є стохастичні моделі, зокрема процеси Маркова, які дозволяють моделювати невизначеність у змінюваних ринкових умовах та можливу зміну поведінки клієнтів, що безпосередньо впливає на обсяги доходів та витрат компанії. Застосування таких моделей забезпечує адаптивність автоматизованої системи, що дозволяє їй працювати у нестабільних умовах і підвищує точність прогнозів, що особливо цінно у випадках раптових змін ринку. Використання стохастичних підходів допомагає не лише виявити закономірності у даних, але й передбачити можливі варіанти розвитку подій, що дозволяє мінімізувати ризики і знизити витрати за умов високої невизначеності. Це значно

підвищує надійність і стійкість фінансової автоматизованої системи до непередбачених зовнішніх змін.

Об'єктно-орієнтовані моделі також роблять значний внесок в автоматизацію бізнес-процесів, оскільки вони дають змогу структурувати всі аспекти діяльності у вигляді системи взаємопов'язаних об'єктів, кожен з яких відповідає за конкретний елемент фінансової діяльності. У сфері управління фінансами такі об'єкти можуть включати в себе компоненти, як-от «Рахунок», «Транзакція», «Платіж» та інші, що безпосередньо стосуються обробки коштів. Кожен об'єкт має свій набір атрибутів і методів, які дозволяють йому взаємодіяти з іншими об'єктами та виконувати конкретні завдання: обробку даних, прогнозування майбутніх витрат, оцінку ризиків, класифікацію та інші фінансові операції. Наприклад, об'єкт «Транзакція» може мати властивості, що дозволяють визначити тип операції (вхідний чи вихідний платіж) і забезпечити цілісність даних при обробці великих масивів фінансової інформації. Використання діаграм класів (наприклад, UML) на етапі проектування допомагає візуалізувати структуру і зв'язки між такими об'єктами, що значно спрощує процес обслуговування та модернізації системи.

Об'єктно-орієнтований підхід також сприяє масштабованості системи. Якщо виникає потреба додати новий модуль для обробки альтернативних типів платежів або впровадити додатковий алгоритм оптимізації, його можна інтегрувати як окремий клас, не вносячи складних змін в існуючу структуру. Це забезпечує модульність і гнучкість автоматизаційної системи, оскільки окремі об'єкти або класи можуть бути легко оновлені або розширені без необхідності переробляти весь код. Наприклад, якщо змінюються вимоги до обробки транзакцій, досить лише модифікувати відповідний клас «Транзакція», що дозволяє швидко адаптувати систему до нових умов, мінімізуючи затрати часу на перепроєктування.

В кінцевому підсумку, інтеграція математичних і об'єктно-орієнтованих моделей у систему автоматизації управління фінансами забезпечує високу точність і адаптивність, дозволяє швидко реагувати на зміни ринкових умов і забезпечує ефективне використання ресурсів, підвищуючи загальну продуктивність і економічну ефективність процесу управління фінансовими потоками. Завдяки

таким підходам, автоматизована система стає надійною і довговічною, оскільки має потенціал для масштабування і модернізації у відповідності до потреб бізнесу, дозволяючи зберігати актуальність і стабільність роботи навіть за умов динамічних змін ринку.

Математичні моделі відіграють ключову роль у формалізації та аналізі бізнес-процесів, забезпечуючи можливість прийняття обґрунтованих рішень на основі точних даних. Використання таких моделей у фінансовій сфері дозволяє ефективно прогнозувати зміни, оптимізувати витрати та підвищувати ефективність управління ресурсами. Наприклад, моделі регресії, зокрема лінійна та багатофакторна, дозволяють аналізувати залежності між різними економічними показниками, оцінюючи їх вплив на результати діяльності. Завдяки цим методам можна будувати прогнози щодо майбутніх доходів і витрат, планувати бюджет і приймати рішення щодо розподілу коштів із урахуванням змін ринкових умов.

Однією з ключових переваг математичних моделей є їх здатність враховувати складність і динамічність бізнес-середовища. Наприклад, стохастичні моделі дозволяють враховувати невизначеності, які часто виникають у фінансових процесах. Такі моделі, як процеси Маркова, допомагають передбачати поведінку клієнтів або динаміку ринків, забезпечуючи адаптивність системи до зовнішніх змін. Вони дозволяють мінімізувати ризики та оптимізувати розподіл ресурсів у ситуаціях високої нестабільності, що є особливо важливим для фінансових систем, які працюють із великими обсягами транзакцій і постійно змінюваними умовами.

Об'єктно-орієнтовані моделі надають структурний підхід до опису бізнес-процесів, представляючи їх як взаємодію об'єктів із певними атрибутами та методами. У фінансовій сфері такі об'єкти, як "Клієнт", "Рахунок" або "Транзакція", дозволяють моделювати складні системи управління фінансами, зберігаючи гнучкість і масштабованість. Наприклад, об'єкт "Транзакція" може включати властивості, що описують тип операції, суму, дату та статус, а також методи, які забезпечують перевірку цілісності даних або обчислення необхідних показників. Завдяки цьому підходу кожен елемент системи можна вдосконалювати чи замінювати без шкоди для загальної структури.

Використання об'єктно-орієнтованого моделювання також сприяє модульності системи, що дозволяє легко інтегрувати нові функціональні можливості. Наприклад, якщо з'являється потреба в автоматизації нового типу платежів, можна розробити окремий модуль, який взаємодітиме з існуючими компонентами без значних змін у кодї. Це забезпечує високу гнучкість системи та її здатність швидко адаптуватися до змін у вимогах бізнесу. Діаграми класів і послідовностей, які широко використовуються на етапі проектування, дозволяють візуалізувати взаємодію між компонентами системи, полегшуючи її розвиток і обслуговування.

Комбінація математичних і об'єктно-орієнтованих підходів створює потужну основу для автоматизації управління фінансовими потоками. Математичні моделі забезпечують точність і обґрунтованість прийнятих рішень, тоді як об'єктно-орієнтовані підходи сприяють структуризації процесів і забезпечують легкість впровадження змін. Разом вони дозволяють створювати інтегровані системи, які здатні адаптуватися до змін і залишатися ефективними навіть за умов динамічного розвитку бізнесу. Такий підхід забезпечує довговічність системи, дозволяючи їй розширюватися й удосконалюватися з мінімальними затратами ресурсів.

Стохастичні моделі, зокрема, сприяють підвищенню надійності системи, особливо у випадках роботи з нестабільними ринковими умовами або великими обсягами даних. Вони дозволяють системам передбачати не лише найімовірніші сценарії, а й враховувати рідкісні події, які можуть суттєво вплинути на результати. Це важливо для фінансових організацій, що прагнуть мінімізувати ризики та зберегти стабільність у своїй діяльності. Завдяки таким підходам автоматизація бізнес-процесів стає не лише інструментом оптимізації, але й стратегічною перевагою, що забезпечує конкурентоспроможність і розвиток.

2.3 Методи навчання та адаптації моделей машинного навчання

Класичні методи машинного навчання пропонують кілька основних підходів до навчання, кожен з яких має свої переваги і недоліки залежно від специфіки

завдання. Навчання з учителем навчання є одним із найбільш поширених методів, який використовується для розв'язання задач, де доступні розмічені дані. У цьому підході модель навчається на наборах даних із відомими вхідними значеннями і відповідними мітками результату, що дозволяє моделі мінімізувати похибки прогнозу. При цьому методи оптимізації, такі як градієнтний спуск, коригують параметри для досягнення точного результату. Навчання з учителем навчання включає різні алгоритми, як-от лінійну регресію для прогнозування та дерева рішень для задач класифікації.

Навчання без учителя навчання, на відміну від навчання з учителем, працює із даними без явних міток, і часто застосовується для кластеризації або виявлення структур у наборі даних. Моделі цього типу корисні для задач, де потрібно знайти приховані закономірності або поділити дані на групи, як це робить алгоритм k-середніх. У випадках, коли розмічені дані обмежені, використовується часткове навчання з вчителем, що дозволяє поєднувати переваги навчання з вчителем і навчання без вчителя, підвищуючи якість класифікації або кластеризації.

Ще один підхід, навчання з підкріпленням, дозволяє моделі приймати рішення на основі зворотного зв'язку від середовища, максимізуючи кумулятивну винагороду. Хоча навчання з підкріпленням частіше застосовується у завданнях управління або в ігрових моделях, його можна використовувати і в фінансовому аналізі, де адаптація до змін навколишнього середовища є критично важливою.

Методи адаптації класичних моделей машинного навчання також дозволяють підвищити їхню гнучкість і стійкість до нових умов. Ітеративне донавчання є одним із таких підходів, що дозволяє моделі поступово оновлюватися новими даними, не потребуючи повного повторного навчання. Це підвищує ефективність використання обчислювальних ресурсів і дозволяє моделі швидко реагувати на зміни. Для боротьби з перенавчанням часто застосовуються методи регуляризації, зокрема L1- і L2-регуляризація, які знижують ризик надмірної складності моделі, забезпечуючи стабільність навіть на невеликих вибірках даних.

Щоб оптимізувати навчання, використовується динамічне регулювання швидкості навчання, яке дозволяє поступово знижувати кроки корекції параметрів,

підвищуючи точність. Крім того, для покращення точності прогнозів часто застосовуються ансамбльні методи, такі як беггінг і бустинг. Ці методи об'єднують результати кількох моделей, як це відбувається у випадку з Random Forests або Gradient Boosting Machines (GBM), що дозволяє уникати помилок окремих моделей.

Перенесення навчання є ще одним важливим підходом, який передбачає використання попередньо здобутих знань моделі для нових завдань. Це дозволяє зменшити потребу в навчанні на великих обсягах даних і є особливо корисним для задач, де дані обмежені або отримати їх є складно.

Кожен із цих методів сприяє покращенню гнучкості, точності та адаптивності моделей, що особливо важливо для фінансових бізнес-систем, де автоматизація потребує швидкого реагування на зміни умов середовища та доступу до нових даних.

Навчання нейронних мереж базується на різних методах оптимізації та адаптації, що сприяють побудові високоточних і стійких моделей для вирішення складних задач. Класичний метод навчання для нейронних мереж — градієнтний спуск, що є ітераційним алгоритмом, який мінімізує функцію втрат через поступове оновлення ваг мережі. На основі розрахунку похідної функції втрат щодо ваг, алгоритм обирає напрямок зміни параметрів, що знижує помилку моделі. Градієнтний спуск має кілька варіацій, зокрема, стохастичний градієнтний спуск (SGD) та міні-пакетний градієнтний спуск, які застосовуються залежно від обсягу даних і ресурсів.

Щоб адаптувати нейронну мережу до умов, де початкове значення градієнта може бути дуже великим чи малим, використовуються адаптивні методи оптимізації, як-от AdaGrad, RMSprop і Adam. Ці методи динамічно коригують швидкість навчання на основі історії градієнтів, що допомагає уникати застою в локальних мінімумах функції втрат і покращує стабільність моделі під час навчання. Особливо Adam є популярним у практиці завдяки своїй здатності поєднувати адаптивну швидкість навчання з моментом, що сприяє швидкій і стабільній конвергенції навіть у великих мережах.

Адаптація нейронних мереж також часто включає використання технік регуляризації, таких як Dropout і Batch Normalization. Dropout знижує ризик перенавчання, випадковим чином відключаючи частину нейронів під час навчання, що змушує мережу узагальнювати знання, не спираючись на окремі параметри. Batch Normalization, у свою чергу, стабілізує розподіл вихідних значень нейронів, нормалізуючи їх на рівні кожного шару. Це дозволяє зменшити проблему вибухаючих і затухаючих градієнтів, що особливо актуально для глибоких мереж.

Для забезпечення адаптивності в умовах зміни вхідних даних широко застосовується метод донавчання, або fine-tuning. Це дозволяє моделі не лише враховувати нові зразки, але й уникати потреби в повному перенавчанні мережі. Fine-tuning є ефективним при роботі з великими пре-тренуваними моделями, де базова архітектура зберігається, а кінцеві шари мережі адаптуються під нові дані. Цей підхід є особливо ефективним для завдань фінансового аналізу, де розподіл даних може змінюватися з часом.

Метод перенесення навчання також має важливе значення для адаптації нейронних мереж до нових задач, дозволяючи застосовувати знання, набуті на одній вибірці, для іншої. Наприклад, при розв'язанні задач розпізнавання патернів у фінансових транзакціях можна використати перенесення моделей з суміжних доменів, що дозволяє скоротити час на навчання та покращити точність навіть за обмеженого набору специфічних даних.

Нейронні мережі також використовують гіперпараметричну оптимізацію, наприклад, через методи Grid Search чи Random Search, що дозволяє визначити найкращі значення таких параметрів, як кількість шарів, швидкість навчання та кількість нейронів у кожному шарі. Це допомагає забезпечити належну адаптацію моделі до конкретної задачі, що є критично важливим для підвищення якості прогнозів і ефективності навчання.

Методи машинного навчання демонструють величезний потенціал у фінансовій сфері, особливо у задачах автоматизації складних процесів, які потребують аналізу великих обсягів даних. Їх ефективність обумовлена здатністю швидко адаптуватися до нових умов, мінімізувати ризик людських помилок та

надавати прогнози з високою точністю. Наприклад, навчання з вчителем дозволяє створювати моделі, що автоматично класифікують транзакції або призначають платежі до відповідних рахунків. Це значно знижує навантаження на співробітників, які займаються ручною обробкою даних, та забезпечує високу швидкість виконання операцій.

У тих випадках, коли доступ до розмічених даних обмежений, корисними стають підходи з навчанням без вчителя, які дозволяють знаходити приховані закономірності у даних. Наприклад, такі алгоритми можуть бути використані для кластеризації клієнтів за поведінковими характеристиками, що допомагає створювати персоналізовані пропозиції або визначати аномальні дії у фінансових транзакціях. Цей підхід є основою для впровадження систем виявлення шахрайства, які дозволяють оперативно реагувати на потенційно підозрілі дії.

Іншим важливим підходом є частково навчання з вчителем, яке поєднує переваги навчання з вчителем і навчання без вчителя моделей. Використовуючи невелику кількість розмічених даних разом із великими обсягами нерозмічених, цей підхід підвищує точність моделі без значних затрат на створення розмічених вибірок. Така методика особливо корисна у фінансових задачах, де доступ до якісно розмічених даних може бути обмеженим.

Навчання з підкріпленням є ще одним перспективним методом, який дозволяє моделі приймати оптимальні рішення в умовах динамічного середовища. У фінансах цей підхід може бути застосований для управління інвестиційними портфелями, де модель вчиться максимізувати дохід, враховуючи ризики та змінність ринкових умов. Такий підхід забезпечує більшу адаптивність і є особливо корисним у довгострокових фінансових стратегіях.

Особливу увагу слід приділяти адаптації моделей до нових умов, що є критично важливим для фінансових систем, які постійно змінюються. Ітеративне донавчання дозволяє моделям оновлювати свої знання на основі нових даних без потреби у повному перенавчанні. Це забезпечує економію ресурсів і прискорює процес інтеграції змін до вже існуючих систем. Регуляризація, у свою чергу,

допомагає уникати перенавчання, зберігаючи стабільність і точність моделей навіть у випадках роботи з обмеженими вибірками даних.

Нейронні мережі є одним із найбільш універсальних інструментів машинного навчання, оскільки вони здатні моделювати складні залежності у великих масивах даних. У фінансових системах вони можуть бути використані для прогнозування поведінки клієнтів, аналізу ризиків або автоматизації задач розподілу коштів. Гнучкість нейронних мереж дозволяє їх адаптувати під різноманітні завдання шляхом налаштування архітектури та гіперпараметрів. Наприклад, перенесення навчання дозволяє використовувати вже навчені моделі для нових задач, що скорочує час розробки та підвищує точність навіть за обмеженої кількості даних.

Додавши до цього системний підхід до інтеграції моделей машинного навчання у бізнес-процеси, можна досягти значних переваг. Мікросервісна архітектура дозволяє впроваджувати окремі компоненти, які виконують спеціалізовані функції, такі як обробка даних, прогнозування чи автоматизація транзакцій. Це забезпечує гнучкість системи та можливість масштабування без необхідності глобальних змін у вже існуючій інфраструктурі. Інтеграція моделей через API спрощує взаємодію між різними елементами системи, що дозволяє бізнесу швидко реагувати на нові виклики та можливості.

Враховуючи динамічний характер фінансових ринків і зростаючу складність бізнес-процесів, впровадження моделей машинного навчання забезпечує не лише автоматизацію, а й стратегічні переваги для компаній. Це дозволяє знижувати витрати, підвищувати точність прийняття рішень і швидкість обробки даних, що у підсумку сприяє довгостроковому зростанню та конкурентоспроможності бізнесу.

2.4 Висновки

Автоматизація бізнес-процесів за допомогою машинного навчання стає важливим інструментом для підвищення ефективності роботи компаній, особливо в таких сферах, як фінансовий менеджмент. Використання алгоритмів машинного навчання дозволяє автоматизувати обробку платіжних транзакцій, пропонувати

оптимальні стратегії розподілу коштів та навіть прогнозувати майбутні фінансові потреби компаній. Це знижує ймовірність людських помилок, що виникають під час виконання рутинних операцій, і прискорює процеси обробки даних. Наприклад, в фінансових системах, де необхідно обробляти великі обсяги платіжних інвойсів, система на основі машинного навчання може визначити найбільш відповідні інвойси для кожного платежу, зменшуючи час, витрачений на виконання цих операцій. Мікросервісна архітектура надає ще одну важливу перевагу для автоматизації бізнес-процесів, дозволяючи створювати гнучкі й масштабовані системи. Вона дозволяє кожному сервісу мати чітко визначену функцію, що знижує складність масштабування та модифікації системи. Застосування цієї архітектури в інтеграції машинного навчання дозволяє легко впроваджувати нові моделі без зміни основної структури системи. Це важливо, оскільки дозволяє швидко адаптуватися до нових вимог бізнесу або технологічних змін без значних витрат на переписування існуючого коду. Інтеграція машинного навчання в бізнес-системи має великий потенціал для подальших досліджень і вдосконалення. Дослідження можуть охоплювати вдосконалення існуючих моделей, створення нових алгоритмів для глибшого аналізу фінансових даних, а також розширення функціональності для забезпечення більш гнучкого управління ресурсами. Наприклад, прогнози на основі великих даних можуть значно полегшити управління фінансами, дозволяючи більш точно передбачати майбутні витрати або доходи, а також пропонувати найбільш ефективні стратегії для їх управління.

Практичне застосування машинного навчання для автоматизації фінансових процесів вже має реальні результати. Системи, які обробляють фінансові транзакції на основі машинного навчання, значно зменшують час обробки і покращують точність. Такі системи можуть автоматично визначати найкращі інвойси для покриття платежів, що допомагає бізнесу працювати більш ефективно і зменшує кількість помилок, пов'язаних з ручним введенням даних. Крім того, автоматизація цих процесів дозволяє забезпечити більшу прозорість у фінансових операціях, що є важливим аспектом для компаній, що працюють із великими обсягами даних і потребують високої точності в управлінні фінансами.

Загалом, застосування машинного навчання в автоматизації бізнес-процесів дозволяє не лише підвищити ефективність роботи, але й створити більш адаптивні та гнучкі системи. Це дає змогу компаніям швидше реагувати на зміни на ринку, оптимізувати ресурси і мінімізувати витрати, що є критичним для успіху в сучасному конкурентному середовищі.

3 АРХІТЕКТУРА ПРОГРАМНОЇ РЕАЛІЗАЦІЇ

3.1 Формування та аналіз вимог програмної реалізації системи

Наша система призначена для автоматизації бізнес-процесів з акцентом на управління фінансами, де особливу роль відіграє процес обробки клієнтських оплат і їх прив'язки до конкретних інвойсів. У поточній системі, коли клієнт вносить оплату, немає можливості точно визначити, який саме інвойс вона покриває. Це створює суттєві труднощі в ручному розподілі коштів, який потребує багато часу і має ризик людських помилок.

Ця проблема має дві основні складові: по-перше, складність у визначенні інвойсу або кількох інвойсів, які має покрити надана сума, по-друге, часті випадки, коли оплата одного клієнта може відповідати не одному, а декільком інвойсам, що потребує додаткової точності при їх співставленні. Тому критичною вимогою до автоматизації цього процесу є висока точність прогнозування, яка дозволить з мінімальними помилками визначити належні інвойси для кожного окремого платежу.

Використання алгоритмів машинного навчання дозволяє вирішити цю проблему, автоматизуючи процес розподілу платежів між інвойсами. Машинне навчання надає можливість аналізувати історичні дані про транзакції, враховувати специфічні патерни та закономірності в поведінці платежів і підбирати оптимальні моделі для прогнозування зв'язку між платіжними сумами та інвойсами. Завдяки цьому наш підхід не тільки зменшує навантаження на співробітників, але й мінімізує ризик помилок, які можуть виникати при ручному розподілі коштів.

Задача автоматизації розподілу платежів між інвойсами передбачає кілька важливих аспектів, які повинні бути враховані при розробці та впровадженні машинного навчання для цієї задачі.

Функціональні вимоги:

– Багатокласова класифікація – для кожного платежу модель повинна визначити, які саме інвойси покриваються цим платіжним внеском.

– Можливість покриття кількох інвойсів одночасно – один платіж може бути розподілений на кілька інвойсів, тому модель повинна вказати всі інвойси, які частково чи повністю покриваються платежем.

– Зниження ручної обробки – автоматизація розподілу платежів дозволить скоротити час та зусилля, необхідні для обробки платежів вручну.

– Простота інтеграції з існуючими системами – система повинна без проблем взаємодіяти з поточною архітектурою бази даних та фінансовими системами компанії.

Нефункціональні вимоги:

– Швидкість обробки – система повинна бути здатною швидко обробляти великі обсяги платежів, без значних затримок.

– Масштабованість – рішення повинно бути здатним підтримувати зростання кількості клієнтів і кількості інвойсів, гарантуючи ефективну обробку без зниження якості.

– Точність прогнозів – модель повинна давати точні рекомендації щодо розподілу коштів, щоб мінімізувати можливість помилок при класифікації платежів.

– Надійність – модель повинна працювати стабільно, без значних збоїв або помилок в умовах різних варіантів платіжних сценаріїв.

Очікувані результати від інтеграції машинного навчання:

– Зменшення кількості помилок при розподілі платежів між інвойсами, що підвищить точність фінансових операцій.

– Підвищення ефективності – автоматизація дозволить значно зменшити час обробки платежів та знизити потребу в ручному втручанні.

– Покращення точності класифікації інвойсів, що дозволить здійснювати фінансові транзакції з високою впевненістю в правильності розподілу платежів.

– Оптимізація ресурсоемних процесів – швидка і точна автоматизація дозволить зекономити ресурси, такі як час працівників та енергетичні витрати на обробку даних.

Таким чином, інтеграція машинного навчання дозволить значно полегшити процес управління фінансовими транзакціями та покращити точність і ефективність системи в цілому.

3.2 Вибір та обґрунтування моделі машинного навчання

Ми вибрали нейронну мережу для вирішення поставленої задачі, оскільки цей тип моделі є ефективним для багатокласових класифікаційних задач, де потрібно враховувати складні, нелінійні взаємозв'язки між входами. У нашому випадку, задачею є автоматизація процесу розподілу платежів за інвойсами, де один платіж може покривати один або кілька інвойсів одночасно. Кожен інвойс має свій набір характеристик, і модель повинна вміти ефективно прогнозувати, які саме інвойси покриває платіж.

Нейронні мережі, і зокрема багат шарові перцептрони (MLP), підходять для таких задач завдяки своїй здатності моделювати складні залежності між входами, що дозволяє враховувати різноманітні фактори: суму платежу, бюджет користувача, час від останнього платежу, а також інші характеристики користувача та інвойсу. Моделі типу MLP здатні працювати з великими обсягами даних і вивчати нелінійні залежності, що є важливим у нашій задачі, оскільки прості лінійні моделі не зможуть ефективно вирішувати таку задачу.

Ми вибрали MLP, оскільки цей тип нейронної мережі оптимально підходить для вирішення задач, де потрібно враховувати багатофакторні зв'язки і працювати з числовими даними, як у нашому випадку. Для цієї задачі архітектура моделі передбачає кілька шарів, що дозволяє ефективно обробляти інформацію та робити прогнози на основі складних взаємозв'язків між характеристиками.

Архітектура обраної моделі виглядає таким чином:

– Вхідний шар: містить нейрони, що відповідають за вхідні дані — сума платежу, бюджет користувача, ідентифікатор користувача та час від останнього платежу.

– Приховані шари: кілька шарів з нейронами, які дозволяють моделі вивчати складні залежності між ознаками. Кількість прихованих шарів і нейронів у кожному з них визначається експериментально.

– Вихідний шар: кожен нейрон відповідає за ймовірність того, чи покривається конкретний інвойс цим платежем. Кількість нейронів у вихідному шарі дорівнює кількості інвойсів, що дозволяє моделі передбачити ймовірність покриття кожного з них.

Модель на базі MLP має кілька важливих переваг. По-перше, вона може обробляти великі обсяги даних і ефективно враховувати складні взаємозв'язки між ними. Це дозволяє досягати високої точності прогнозів навіть при великій кількості вхідних ознак. По-друге, нейронні мережі мають високу гнучкість і можуть адаптуватися до змін у даних, що є важливим для нашої системи, де інвойси і платежі можуть змінюватися з часом.

З недоліків можна відзначити високу потребу в обчислювальних ресурсах для навчання мережі, особливо при великій кількості вхідних параметрів. Крім того, процес налаштування гіперпараметрів (наприклад, кількість шарів, кількість нейронів у кожному шарі, швидкість навчання тощо) може бути складним і вимагати значного часу на експерименти. Однак ці недоліки компенсуються потужністю моделі в розв'язанні складних класифікаційних задач.

Вибір нейронної мережі дозволяє досягти високої точності в класифікації інвойсів для кожного платежу, що в свою чергу автоматизує процес і зменшує ймовірність помилок при розподілі коштів між інвойсами.

Архітектура цієї мережі буде включати кілька основних компонентів:

Вхідний шар:

– Кількість нейронів у цьому шарі дорівнює кількості вхідних ознак. Для нашої задачі це:

- сума платежу (1 ознака),
- бюджет користувача (1 ознака),
- ідентифікатор користувача (1 ознака),
- час від останнього платежу (1 ознака).

- Усього: 4 нейрони на вхідному шарі.

Приховані шари:

- Мережа буде мати два приховані шари, що дозволяють моделі вивчати складні зв'язки між вхідними ознаками.

– Кількість нейронів у кожному з прихованих шарів буде варіюватися в залежності від експериментальних налаштувань, але для початкового варіанту можна використовувати від 64 до 128 нейронів у кожному шарі. Ці нейрони виконують нелінійні перетворення і допомагають моделі вивчати складні патерни у даних.

– Для активації у цих шарах зазвичай застосовується ReLU (Rectified Linear Unit), оскільки вона дозволяє моделі ефективно навчатися, уникати проблеми затухаючого градієнта та прискорювати навчання. Формула(3.1) цієї активаційної функції:

$$f(x) = \max(0, x) \quad (3.1)$$

де x — це значення, яке обробляється на кожному етапі мережі

Графік функції ReLU показаний на рисунку 3.1.

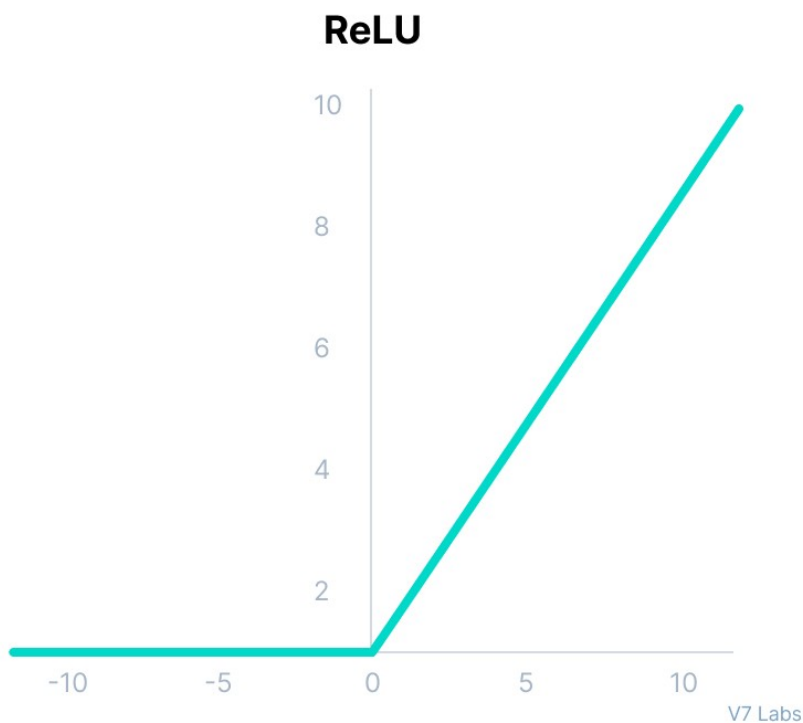


Рисунок 3.1 - Графік функції ReLU

Вихідний шар:

– Вихідний шар складається з нейронів, рівних кількості інвойсів, що повинні бути покриті платежем. Якщо в системі є до 6 типів інвойсів, то в цьому шарі буде 6 нейронів.

– Кожен нейрон на виході буде відповідати ймовірності того, чи покривається відповідний інвойс платіжною сумою.

– Оскільки задача є багатокласовою класифікацією, на виході застосовуємо активаційну функцію sigmoid, формула якої виглядає наступним чином:

$$\sigma(x) = \frac{1}{1+e^{-x}} \quad (3.2)$$

де:

- x — це вхідне значення, яке передається до нейрону.
- e — це основа натурального логарифму (приблизно 2.71828).

Графік функції sigmoid відображений на рисунку 3.2.

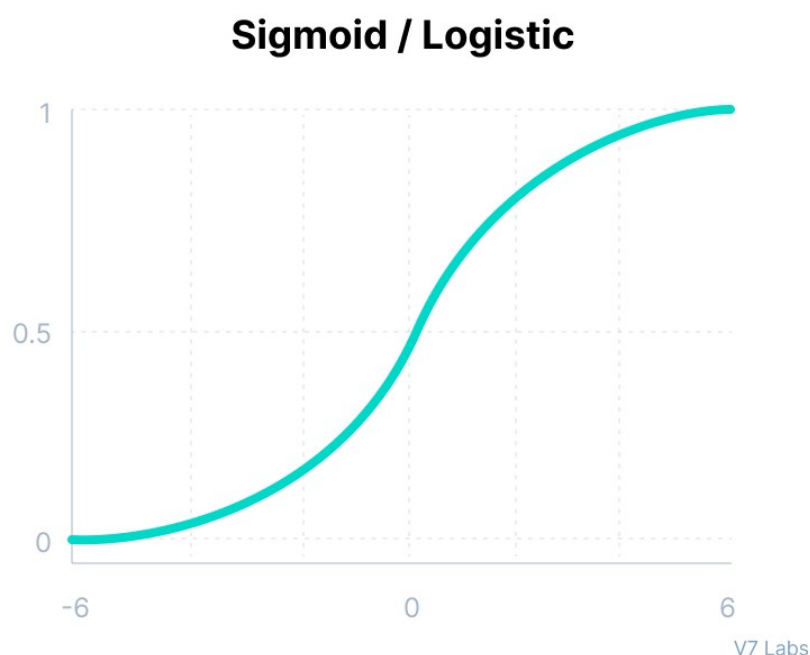


Рисунок 3.2 - графік функції sigmoid

Вона дозволяє кожному нейрону вивести ймовірність того, що платіж покриває інвойс (від 0 до 1). Таким чином, кожен нейрон повертає значення 0 або 1 для кожного інвойсу.

Схематично архітектура нейронної мережі показана на рисунку 3.3

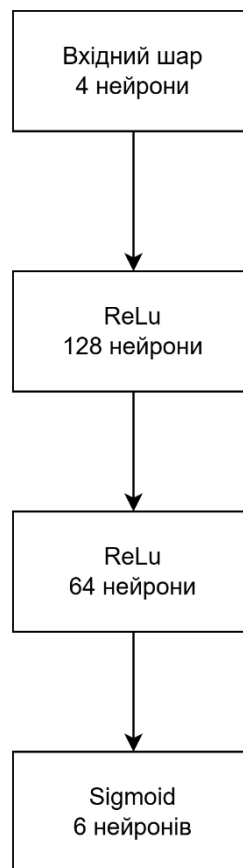


Рисунок 3.3 - Схема архітектури нейронної мережі

Функція втрат:

– Для багатокласової класифікації з багатьма мітками, використовується функція втрат Binary Cross-Entropy, оскільки для кожного класу (інвойсу) ми маємо дві можливості — покрито чи не покрито.

Оптимізація:

– Для оптимізації параметрів мережі буде використовуватися алгоритм Adam (Adaptive Moment Estimation), який є ефективним і часто використовуваним методом для навчання нейронних мереж, оскільки забезпечує швидку і стабільну конвергенцію.

– Для оцінки продуктивності моделі в задачі автоматизованого розподілу платежів по інвойсах використовуються кілька основних метрик. Вони дозволяють отримати багатогранну оцінку ефективності моделі, звертаючи увагу на різні аспекти класифікації.

– Accuracy (Точність класифікації) є основною мірою коректності роботи моделі, яка обчислює частку правильно класифікованих випадків від загальної кількості. Ця метрика виражається формулою:

$$Accuracy = \frac{\text{Загальна кількість інвойсів}}{\text{Кількість правильно класифікованих інвойсів}} \quad (3.3)$$

Accuracy забезпечує загальний показник точності і зручний для базової оцінки моделі, хоча в задачах з незбалансованими даними його може бути недостатньо, оскільки висока точність на домінуючих класах може перекривати неточності на менш представлених.

Precision (Точність) обчислює частку правильних передбачень серед усіх випадків, класифікованих як певний клас. Ця метрика допомагає уникати хибно-позитивних передбачень, тобто випадків, коли оплата зараховується до неправильного інвойсу. Формула Precision:

$$Precision = \frac{TP}{TP + FP} \quad (3.4)$$

де:

- TP – це кількість випадків, коли модель правильно передбачила наявність певного класу,
- FP - це кількість випадків, коли модель неправильно передбачила наявність певного класу.

Precision є важливим показником для задач, де критично уникати неправильних класифікацій, адже знижує ризик некоректної розподіли платежів, що є пріоритетом у нашій системі.

Recall (Повнота) дозволяє оцінити частку коректно передбачених випадків для певного класу серед усіх випадків цього класу в реальних даних. Recall визначається формулою:

$$Recall = \frac{TP}{TP + FN} \quad (3.5)$$

де:

- TP – це кількість випадків, коли модель правильно передбачила наявність певного класу,
- FN - це кількість випадків, коли модель неправильно передбачила відсутність певного класу.

Високий показник Recall вказує на те, що модель ефективно знаходить усі релевантні інвойси для здійснення платежів, зменшуючи ймовірність пропущених відповідних інвойсів.

F1 Score (Збалансована міра точності та повноти) є гармонійним середнім між Precision і Recall, що дозволяє враховувати обидва аспекти одночасно. Ця метрика особливо цінна для задач, де однаково важливо зменшити як хибно-позитивні, так і хибно-негативні результати:

$$F1\ Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (3.6)$$

F1 Score важливий для балансування точності та повноти, особливо коли обидві метрики мають однаковий пріоритет, адже допомагає оптимізувати розподіл платежів між інвойсами і мінімізує помилки обох типів.

Вплив бізнес-логіки на метрики оцінки моделі є важливим аспектом у системі розподілу платежів за інвойсами. Основна задача бізнесу — мінімізувати фінансові ризики та забезпечити максимально точне спрямування коштів. Вибір метрик і їх інтерпретація повинні враховувати специфіку завдання, оскільки помилки різних типів, зокрема хибно-позитивні (FP) та хибно-негативні (FN), мають різний вплив на результати.

Хибно-позитивні помилки, де модель неправильно вказує, що певний інвойс покривається поточною оплатою, можуть призвести до фінансових невідповідностей, таких як надмірне покриття рахунків, що не є очікуваним. Це особливо критично, оскільки помилка FP означає, що кошти можуть бути помилково спрямовані на інший інвойс, залишаючи потрібний інвойс непокритим. Така ситуація потребує подальшого ручного коригування, що знижує ефективність системи та збільшує ризик неповного або неправильного покриття зобов'язань.

Хибно-негативні помилки, навпаки, означають, що система не включила інвойс, який мав би бути покритим. Це також може викликати проблеми, оскільки інвойси, що потребують покриття, залишаються неврахованими, що потенційно погіршує якість обслуговування клієнта. Проте, такі помилки можуть бути менш критичними в системі, де допустима певна варіативність у виборі інвойсів для покриття.

У бізнес-контексті дуже важливим є вибір оптимального співвідношення між Precision та Recall. Наприклад, якщо FP мають більші негативні наслідки, ніж FN, то Precision може мати більший пріоритет. Це означає, що система повинна зосереджуватися на тому, щоб кожен визначений інвойс дійсно відповідав оплаті, навіть якщо це зменшить загальну повноту (Recall). У такому випадку менша кількість інвойсів може залишатися непокритою (FN), але надмірне покриття (FP) буде знижено.

Вплив бізнес-логіки може також реалізовуватись через вагові коефіцієнти для помилок у процесі навчання моделі. Наприклад, щоб зменшити кількість хибно-позитивних помилок, можна застосувати більшу вагу для FP у функції втрат. Такий підхід дозволить моделі краще враховувати реальні потреби бізнесу, покращуючи точність розподілу коштів і зменшуючи кількість коригувань.

У результаті модель адаптується до бізнес-контексту, а метрики відображатимуть реальну ефективність системи з точки зору покриття інвойсів відповідно до бізнес-логіки та вимог до точності.

Отже, у цьому підрозділі ми здійснили обґрунтований вибір моделі для вирішення завдання автоматизації розподілу платежів за інвойсами, враховуючи специфічні вимоги та бізнес-логіку системи. Нейронна мережа обрана як оптимальна модель, здатна ефективно працювати з багатокласовою класифікацією, що дозволяє оцінювати кожен інвойс на предмет його покриття конкретною оплатою. Визначені метрики, зокрема Precision і Recall, відображають специфіку задачі: досягнення максимальної точності у визначенні інвойсів, що дійсно відповідають оплатам, та уникнення надмірного покриття.

Також було враховано вплив бізнес-логіки, яка визначає пріоритети для метрик оцінки та вимагає особливого підходу до функції втрат для оптимізації моделі. Це дозволяє адаптувати модель до практичних потреб, забезпечуючи надійну автоматизацію процесу та мінімізуючи ризики, пов'язані з помилками у розподілі коштів.

3.3 Розробка алгоритмів для автоматизованого розподілу платежів

У цьому підрозділі розглядається розробка алгоритмів автоматизованого розподілу платежів, що поєднує моделі машинного навчання з бізнес-логікою для забезпечення високої точності та ефективності процесу. Ця автоматизація покликана зменшити кількість ручних операцій та підвищити надійність фінансових операцій, а також адаптувати систему до змінних умов.

Перший етап розподілу платежів – це попередня обробка даних. На цьому етапі алгоритм отримує вхідні параметри, такі як сума платежу, бюджет клієнта, ID клієнта, час з останнього платежу тощо. Основне завдання тут – забезпечення коректності й повноти даних шляхом їх нормалізації, очищення та, при необхідності, заповнення відсутніх значень або видалення аномалій. Попередня обробка сприяє адаптації вхідних даних до вимог нейронної мережі, забезпечуючи послідовність та стабільність у процесі їх обробки.

Другий етап – застосування обраної моделі машинного навчання, в нашому випадку, нейронної мережі, для визначення покриття інвойсів. Нейронна мережа була обрана через її здатність до обробки складних зв'язків між даними, що робить її придатною для задачі багатокласової класифікації. На цьому етапі модель повертає для кожного інвойса ймовірності того, що даний платіж може покрити конкретний інвойс, використовуючи значення 1 або 0 для кожного класу.

Третій етап, інтеграція бізнес-логіки, адаптує результати моделі до реальних умов та пріоритетів бізнесу. Це включає впровадження правил, що враховують категорії інвойсів, порядок їх пріоритетності та додаткові критерії, що можуть мати вагоме значення у фінансовій системі, наприклад:

сортування інвойсів за важливістю або відповідними категоріями;
визначення порядку покриття, враховуючи терміновість або фінансову важливість конкретного інвойсу.

Четвертий етап, фінальна перевірка та оптимізація. На цьому етапі алгоритм перевіряє правильність розподілу платежів, відповідність бізнес-правилам та забезпечує інтеграцію результатів у систему. Це запобігає надмірному покриттю інвойсів або випадковим збоям. Застосування цього підходу є обґрунтованим з огляду на необхідність точності та уникнення помилок, що можуть впливати на фінансову цілісність даних.

Завершальний етап – оформлення результатів та їх збереження у базі даних. Алгоритм записує дані про покриті інвойси, зберігає їх для подальшого використання та забезпечує інтеграцію результатів у звітні модулі системи. У випадку виникнення нестандартних ситуацій, система може сигналізувати про необхідність ручного перегляду.

Першим етапом підготовки даних для автоматизованого розподілу платежів є витягування та обробка ключової інформації з існуючих даних системи. Для коректної роботи моделі необхідно отримати точні відомості про останні платежі користувачів, а саме час здійснення останнього платежу.

Це важливо, оскільки час від останнього платежу є однією з ключових фіч, що використовуються для прогнозування й автоматизації розподілу платежів на відповідні інвойси. Для цього потрібно здійснити кілька кроків:

– Витягування даних про платежі. З бази даних необхідно витягнути інформацію про всі здійснені платежі користувачів, включаючи дату та суму кожного платежу. Важливо також зберегти ідентифікатори користувачів, щоб мати змогу відслідковувати їхню платіжну активність.

– Обчислення часу від останнього платежу. Для кожного користувача потрібно визначити час, що минув з моменту його останнього платежу. Це можна зробити шляхом обчислення різниці між поточним часом і датою останнього здійсненого платежу для кожного користувача. Отримане значення є важливою

фічею для моделі, оскільки воно може вказувати на те, чи є користувач активним платником, чи, можливо, затримує платежі.

– Форматування даних. Після отримання та обробки даних про останній платіж і часу від нього, ці дані мають бути підготовлені для подальшого використання в моделі. Наприклад, якщо дані мають бути представлені в числовому вигляді, необхідно перевести час у відповідний формат, наприклад, в кількість днів чи годин, що минули з моменту останнього платежу.

Наступним етапом після підготовки даних є їх нормалізація. Оскільки наші дані містять ознаки з різними масштабами та одиницями вимірювання, необхідно привести їх до єдиного стандарту для забезпечення коректної роботи моделі машинного навчання. У нашому випадку обрано Z-нормалізацію (стандартизацію), оскільки вона дозволяє ефективно обробляти дані з різними розподілами та масштабами, запобігаючи домінуванню ознак із більшими значеннями, таких як сума платежу.

Процес нормалізації включає віднімання середнього значення ознаки та ділення на її стандартне відхилення. Це забезпечує, що кожна ознака матиме середнє значення 0 та стандартне відхилення 1. Формула Z-нормалізації виглядає наступним чином:

$$X_{norm} = \frac{X - \mu}{\sigma} \quad (3.7)$$

де:

- X — значення ознаки,
- μ — середнє значення ознаки по всіх спостереженнях,
- σ — стандартне відхилення ознаки.

Після нормалізації даних наступним кроком є їхній поділ на тренувальний, валідаційний та тестовий набори. Такий поділ необхідний для коректної оцінки моделі на різних етапах, що дозволяє уникнути перенавчання та забезпечити узагальнюючу здатність. Кожен із наборів має свої функції:

– Тренувальний набір. Використовується для безпосереднього навчання моделі та налаштування її вагових коефіцієнтів. Завдяки цьому набору модель засвоює залежності між вхідними ознаками та мітками.

– Валідаційний набір. Дозволяє відстежувати якість моделі під час навчання та підбирати оптимальні гіперпараметри. Валідаційний набір використовується також для оцінки моделі на проміжних етапах, що дає змогу вчасно виявити ознаки перенавчання.

– Тестовий набір. Призначений для фінальної оцінки продуктивності моделі на нових, раніше невідомих даних, що імітують реальні умови її використання. Тестування на цьому наборі дозволяє отримати об'єктивні показники якості та стабільності моделі.

Додатково, у разі значної нерівномірності в частотах класів, слід розглянути методи балансування, такі як зміна вагових коефіцієнтів класів або створення підвбірок, що підвищить ефективність класифікації та допоможе уникнути упередженості моделі до класів із більшою кількістю спостережень.

Після підготовки даних модель переходить до **етапу тренування**, де відбувається процес оптимізації її параметрів для точного передбачення відповідності платежів інвойсам. На цьому етапі мережа навчається на великому обсязі історичних даних, де для кожного запису відомий зв'язок між сумою платежу та інвойсом, який він покриває.

– Розподіл даних. Тренувальний набір розділяється на тренувальну та валідаційну вибірки, щоб забезпечити незалежну оцінку продуктивності моделі під час тренування.

– Функція втрат і оптимізація. Ми використовуємо функцію бінарної крос-ентропії як функцію втрат для багатокласової класифікації. Ця функція оцінює різницю між передбачуваними значеннями й істинними значеннями, зменшуючи її за допомогою методу зворотного поширення помилки (backpropagation) та оптимізатора Adam.

– Процес тренування. Модель навчається протягом кількох епох, на кожній з яких оновлюються ваги нейронів з метою мінімізації функції втрат. Кожна епоха

дозволяє моделі поступово покращувати точність передбачень, підлаштовуючись до особливостей даних.

Цей процес завершується, коли модель досягає стабільної точності на валідаційній вибірці, що свідчить про її готовність до використання для передбачення цільових інвойсів на нових платежах.

Схема на рисунку 3.3 зображує алгоритм обробки даних після тренування моделі машинного навчання, що автоматично розподіляє платежі між інвойсами. Ось детальний опис кожного етапу:

Отримання нових даних: На початковому етапі процесу ми отримуємо нові дані про користувачів та їхні платежі. Це може бути інформація про дату платежу, суму, бюджет користувача тощо.

Попередня обробка даних: На цьому етапі здійснюється очищення та підготовка даних для подальшої обробки. Це може включати видалення пропусків, коригування неточностей або аномалій у даних.

Нормалізація даних: Оскільки дані можуть мати різні масштаби та одиниці вимірювання (наприклад, сума платежу і бюджет користувача), їх нормалізують. Це дозволяє всім ознакам мати однакову "вагу" під час навчання моделі, що покращує її ефективність.

Прогнозування (модель ML): На цьому етапі оброблені та нормалізовані дані використовуються для прогнозування моделі машинного навчання. Модель прогнозує, які інвойси мають бути покриті для кожного платежу.

Аналіз результатів моделі: Після прогнозування модель надає набір результатів, які необхідно проаналізувати. Це може включати оцінку точності прогнозу, аналіз помилок і порівняння з реальними даними.

Визначення інвойсів для покриття: На цьому етапі відбувається прийняття рішення на основі результатів моделі, тобто визначення, який інвойс або інвойси слід покрити за рахунок конкретного платежу. Кожен результат прогнозу має бути зіставлений із реальною ситуацією.

Формування результату та відображення: На заключному етапі формується кінцевий результат: інвойси, що покриваються за допомогою цього платежу.

Результат відображається користувачу або передається на наступний етап в системі.

Ця схема чітко показує послідовність кроків, необхідних для автоматичного розподілу платежів на основі даних, що поступають. Важливо, що кожен етап є логічно послідовним і ґрунтується на попередньому, що забезпечує ефективність і коректність роботи системи.

Алгоритм обробки даних після тренування моделі

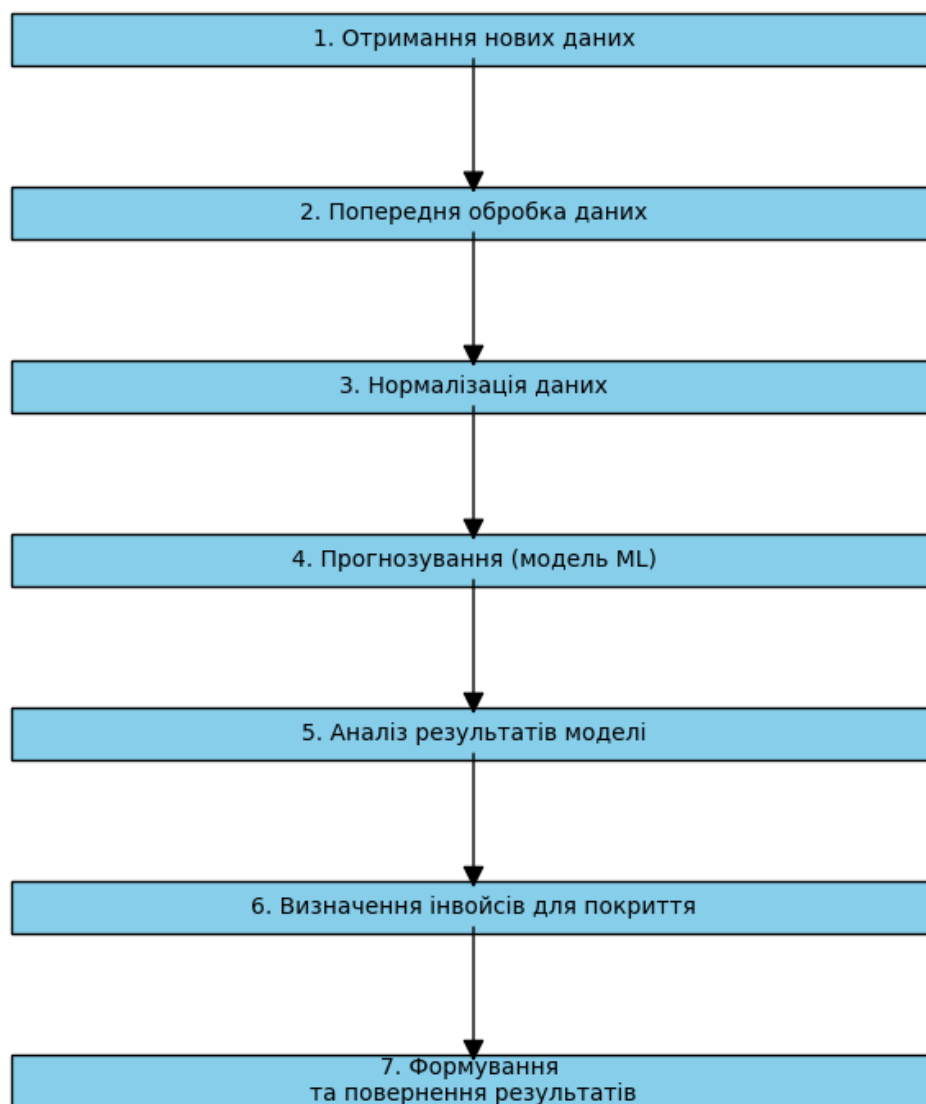


Рисунок 3.3 - Алгоритм обробки платежу та отримання результату

3.4 Архітектурний та компонентний дизайн системи

У рамках системи автоматизованого розподілу платежів наша частина системи відповідає за прогнозування, на основі машинного навчання, покриття інвойсів за допомогою платежів. Архітектура цієї частини системи базується на мікросервісному підході, що дозволяє розподіляти навантаження, забезпечувати масштабованість та зручність обслуговування. Кожен сервіс у цій архітектурі виконує конкретну задачу, що забезпечує високу гнучкість і можливість розширення без порушення роботи інших частин системи.

Основні компоненти нашої мікросервісної частини:

– Мікросервіс для обробки платіжних транзакцій: Цей сервіс відповідає за отримання даних про платіжні транзакції, їхню обробку та підготовку для подальшої класифікації. Мікросервіс працює з платіжними шлюзами або іншими зовнішніми системами для отримання необхідної інформації про платіж: суму, дату, користувача та інші параметри. Він також може містити логіку для перевірки валідності платежу. Після обробки даних, цей мікросервіс передає їх до наступного етапу системи — мікросервісу для класифікації інвойсів.

– Мікросервіс для класифікації інвойсів: Це ключовий мікросервіс нашої системи. Він здійснює прогнозування того, які інвойси будуть покриті даним платежем, використовуючи модель машинного навчання. Мікросервіс отримує оброблені дані про платіж від попереднього сервісу і передає їх на вхід моделі машинного навчання. Після того, як модель робить прогноз (які інвойси слід покрити), мікросервіс передає результати до мікросервісу для взаємодії з базою даних, щоб зберегти інформацію про те, які інвойси були покриті якими платежами.

– Мікросервіс для роботи з базою даних: Цей сервіс відповідає за зберігання, отримання та оновлення даних, пов'язаних з користувачами, платежами та інвойсами. База даних містить таблиці для зберігання інформації про платіжні транзакції, інвойси, їх статуси та інші ключові елементи. На рисунку 3.4 представлено схему даних які мають використовуватись цим мікросервісом.

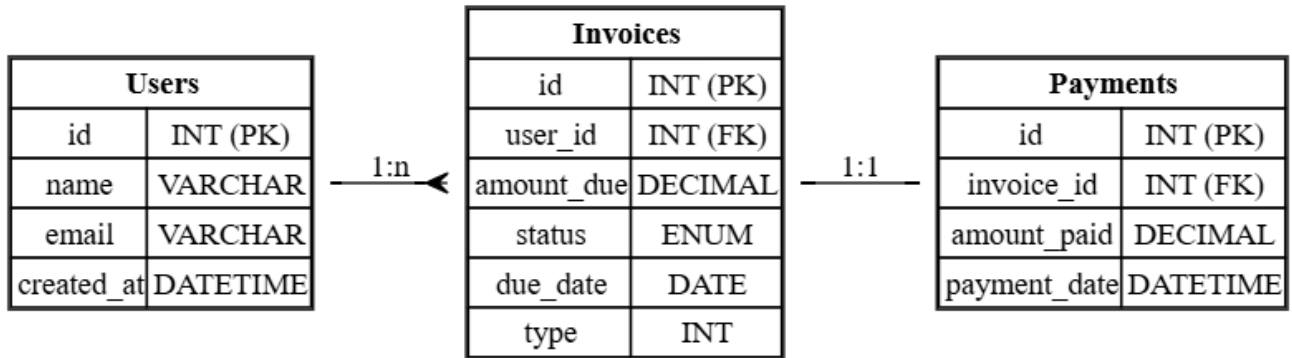


Рисунок 3.4 - Схема даних які використовуються в мікросервісі

Мікросервіс працює як інтерфейс для збереження результатів обробки платежів, таких як зв'язки між платежами та інвойсами. Також він може використовуватись для отримання історичних даних для подальшої обробки та тренування моделі.

– Мікросервіс для дотренування моделі : У разі, коли обсяги даних зростають і необхідно покращити точність моделі, ми додаємо ще один важливий компонент — мікросервіс для дотреноування моделі машинного навчання. Цей мікросервіс забезпечує безперервне вдосконалення моделі, дозволяючи системі адаптуватися до нових даних та підтримувати актуальність її прогнозів. Обробка нових даних: Після збору нових даних мікросервіс ініціює процес їх обробки та нормалізації, що дозволяє привести дані до єдиного формату та масштабу, який підходить для подачі в модель машинного навчання. Для цього застосовуються ті ж самі техніки обробки, що й на етапі початкового тренування моделі, такі як Z-нормалізація. Дотреноування моделі: Мікросервіс використовує нові дані для дотреноування вже існуючої моделі. Важливо зазначити, що це не є повним перенавчанням моделі, а лише корекцією з урахуванням нових даних. Дотреноування моделі відбувається з використанням невеликого обсягу даних, що дозволяє швидко адаптувати модель до нових змін, не потребуючи значних обчислювальних ресурсів. Оцінка моделі: Після кожного етапу дотреноування мікросервіс проводить оцінку моделі на основі метрик, таких як точність, precision, recall та F1-score. Це дозволяє перевірити, чи покращились прогнози моделі після навчання на нових даних. Якщо модель демонструє значні покращення, вона замінюється на нову версію, і тільки після цього знову починається її використання

в процесі класифікації інвойсів. Оновлення моделі в продуктивному середовищі: Після того як модель пройшла процес дотреновування та отримала задовільні результати, вона оновлюється у продуктивному середовищі. Мікросервіс для дотреновування моделі також може бути відповідальним за створення нової версії моделі і передачу її в основну систему для подальшого використання.

– API-шлюз: API-шлюз виступає як точка взаємодії з іншими компонентами системи, зокрема з користувацьким інтерфейсом та іншими системами, що можуть бути інтегровані в майбутньому. Він отримує запити від користувачів, які хочуть переглянути свої платежі, інвойси та статуси покриття, і передає ці запити відповідним мікросервісам. API-шлюз також відповідає за маршрутизацію запитів до відповідних мікросервісів та повернення результатів.

Всі ці мікросервіси взаємодіють між собою за допомогою REST API або інших комунікаційних протоколів. Платіжні дані, що надходять до мікросервісу обробки платежів, потім передаються до мікросервісу для класифікації, який здійснює прогноз і передає його до мікросервісу для роботи з базою даних, де зберігаються результати. Весь процес контролюється API-шлюзом, що дозволяє здійснювати інтеграцію з іншими частинами системи та забезпечує комунікацію з кінцевим користувачем.

У випадку коли ми збільшили обсяг даних то запускаємо мікросервіс для дотреновування моделі Архітектура мікросервісу для дотреновування **моделі**

– Запуск збору нових даних — цей компонент відповідає за моніторинг і збір нових платіжних транзакцій, що надходять у систему.

– Обробка даних — цей мікросервіс виконує попередню обробку, очистку та нормалізацію нових даних перед тим, як вони будуть подані для дотреновування моделі.

– Процес навчання — основна логіка для додаткового тренування моделі з новими даними, яка запускається тільки в разі потреби.

– Оцінка моделі — після кожного циклу навчання мікросервіс перевіряє, чи покращилися результати на основі метрик.

– Оновлення моделі — якщо модель має покращення, вона замінюється новою версією для використання в основному процесі класифікації інвойсів.

Схема роботи цього мікросервісу показана на рисунку 3.5

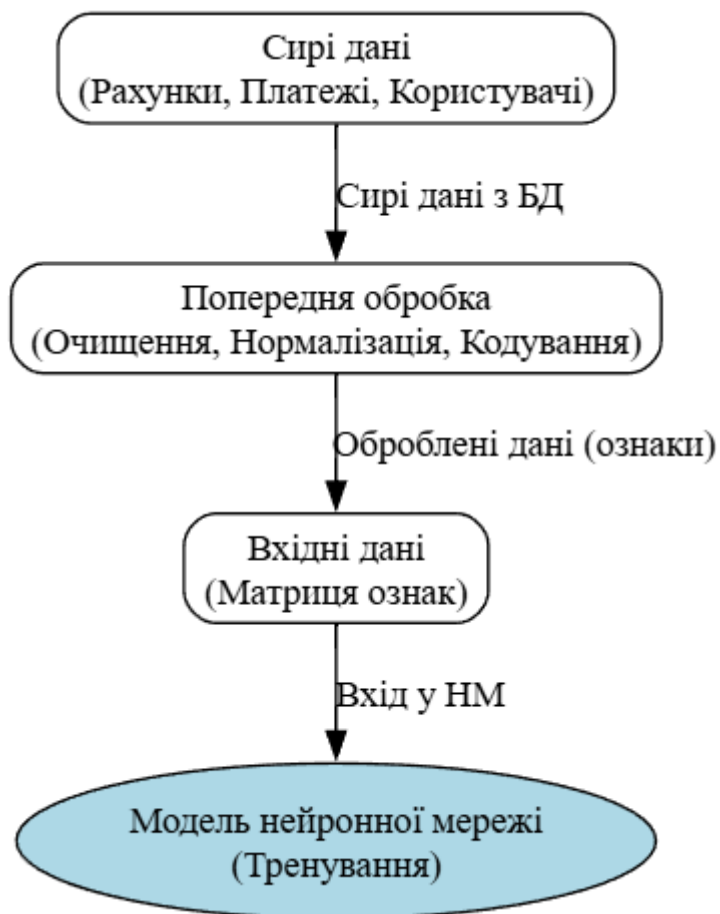


Рисунок 3.5 - Схема роботи мікросервісу тренування мережі

Ця архітектура дозволяє побудувати незалежні, але інтегровані компоненти, що забезпечують високий рівень масштабованості, стійкості та гнучкості для подальших оновлень або змін. Модуль машинного навчання може бути оновлений або замінений без впливу на інші компоненти, а база даних може бути змінена або замінена без порушення роботи решти системи.

Таким чином, наша мікросервісна архітектура забезпечує ефективне, масштабоване та гнучке рішення для автоматизованого розподілу платежів між інвойсами з використанням машинного навчання. Схема взаємодії мікросервісів показана на рисунку 3.4

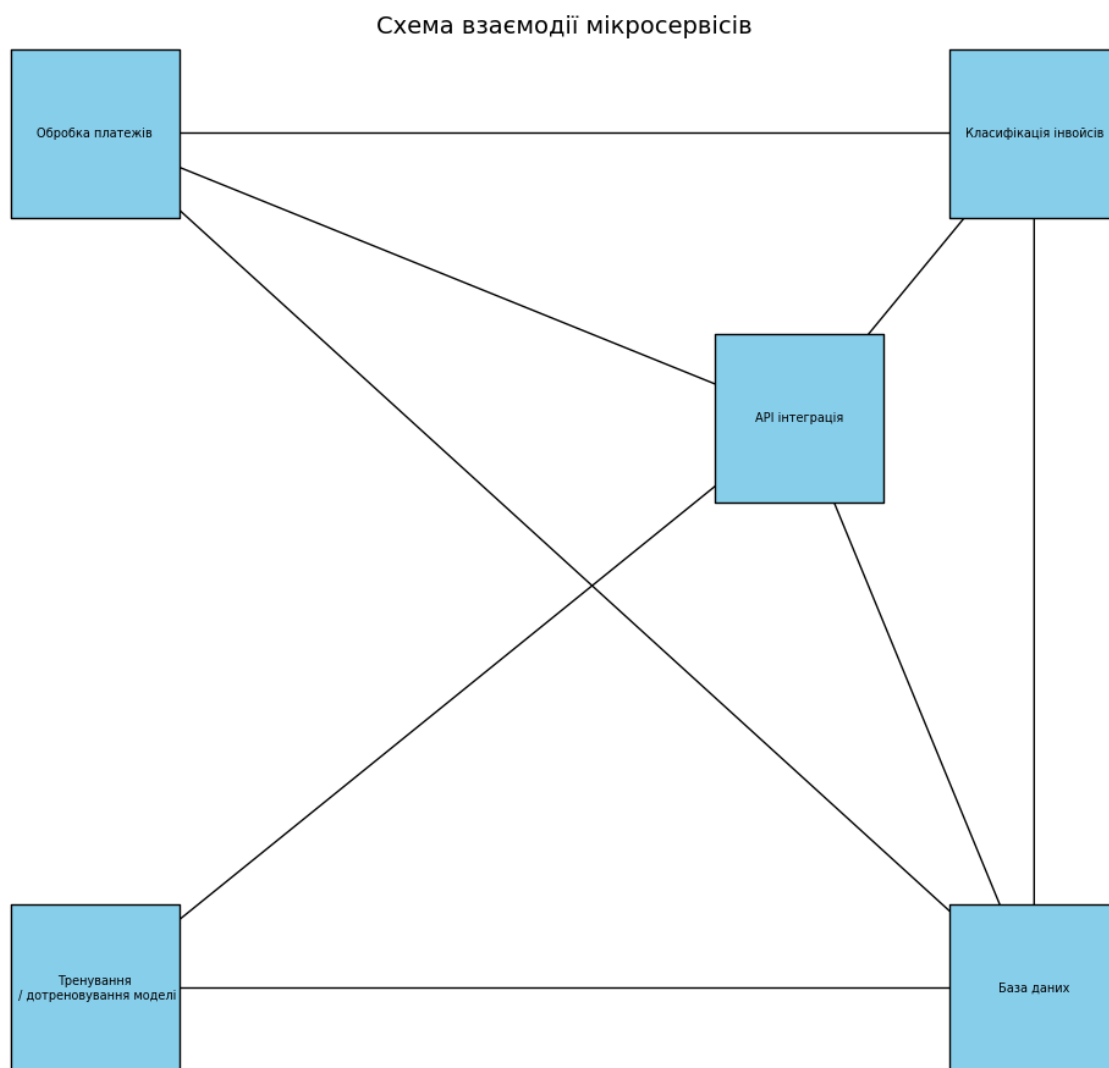


Рисунок 3.4 - Схема взаємодії мікросервісів

3.5 Технічні вимоги до програмних та апаратних засобів

Для забезпечення коректної роботи моделі в реальному часі, а також для підтримки необхідної швидкості обробки даних і надійного зберігання, визначено такі програмні та апаратні вимоги. Враховано обсяг і тип даних, що обробляються, передбачуване навантаження на систему, необхідність масштабованості та безперервного доступу до моделі для оптимального користувацького досвіду.

Програмні вимоги:

– Операційна система: Linux для серверної частини або хмарні сервіси (AWS, Google Cloud) із підтримкою контейнерів.

- Мова програмування: Python як основна мова розробки з використанням бібліотек TensorFlow, NumPy та Pandas.

Апаратні вимоги:

- Процесор (CPU): Багатоядерний процесор (мінімум 8 ядер) для швидкої обробки даних.

- Графічний процесор (GPU): GPU для задач машинного навчання та швидкого дотренування моделі.

- Оперативна пам'ять (RAM): Мінімум 32 ГБ для ефективного завантаження та обробки даних.

- Система зберігання даних: SSD-диски об'ємом від 1 ТБ для швидкого доступу до бази даних і зберігання результатів.

Вимоги для обробки великих обсягів даних та масштабованості:

- Контейнеризація: Використання Docker для легкої масштабованості мікросервісів.

- Оркестрація контейнерів: Kubernetes для автоматичного масштабування та балансування навантаження.

- Моніторинг та логування: Використання Prometheus та Grafana для моніторингу продуктивності системи.

- Підтримка масштабованості:

- Передбачено гнучке масштабування серверних ресурсів для обробки зростаючих обсягів даних.

- Використання хмарних обчислень для автоматичного розгортання додаткових екземплярів мікросервісів.

3.6 Висновок

У процесі розробки системи автоматизованого розподілу платежів між інвойсами на основі машинного навчання було здійснено кілька ключових етапів, включаючи формування вимог, вибір моделі машинного навчання, проектування архітектури системи, технічні вимоги та характеристику взаємодії між мікросервісами.

Для розробки системи було визначено основні вимоги до її функціональності. Зокрема, система повинна автоматизувати процес розподілу платежів між інвойсами, забезпечувати високу точність прогнозів і підтримувати безперервний доступ до системи. Важливим аспектом є адаптація системи до нових даних для підтримки актуальності прогнозів, а також надання зручного інтерфейсу для перегляду платіжних транзакцій та статусів інвойсів.

Для вирішення задачі прогнозування покриття інвойсів було обрано алгоритми машинного навчання, зокрема методи класифікації, які дозволяють на основі платіжних даних здійснювати прогноз, які інвойси повинні бути покриті цим платежем. Вибір конкретної моделі був обумовлений необхідністю обробки складних даних, таких як різноманітні платежі та інвойси з різними параметрами. У результаті було прийнято рішення використовувати класичні моделі машинного навчання, такі як логістична регресія або дерева рішень, з подальшою можливістю донавчання моделі для підвищення точності її прогнозів.

Архітектура системи базується на мікросервісному підході, що забезпечує розподілену обробку даних, масштабованість та високий рівень доступності. Кожен мікросервіс виконує певну задачу, що забезпечує гнучкість і можливість масштабування. Основними компонентами системи є: мікросервіс для обробки платіжних транзакцій, мікросервіс для класифікації інвойсів, мікросервіс для роботи з базою даних, мікросервіс для дотреновування моделі та API-шлюз для взаємодії з користувачем. Кожен із цих компонентів виконує конкретну роль у процесі автоматизації, що дозволяє оптимізувати та спростити роботу всієї системи.

Для забезпечення ефективної роботи системи були визначені технічні вимоги до програмного та апаратного забезпечення. Система повинна працювати на операційній системі Linux або в хмарному середовищі (AWS, Google Cloud), з використанням контейнеризації для забезпечення масштабованості. Мова програмування Python була вибрана через її потужні бібліотеки для машинного навчання, такі як TensorFlow, NumPy та Pandas. Щодо апаратного забезпечення, потрібен багатоядерний процесор (мінімум 8 ядер), графічний процесор (GPU) для

прискороного навчання моделей, мінімум 32 ГБ оперативної пам'яті та SSD-диски для швидкого зберігання даних.

Однією з ключових частин системи є процес дотреновування моделі. З кожним новим набором платіжних даних необхідно покращувати точність прогнозів. Для цього був розроблений мікросервіс для дотреновування моделі, який здійснює збір нових даних, їх попередню обробку та нормалізацію, а потім навчання моделі. Після кожного циклу навчання модель оцінюється за допомогою метрик точності, precision, recall та F1-score. Якщо модель демонструє поліпшення, вона оновлюється в продуктивному середовищі.

Всі компоненти системи взаємодіють між собою через REST API або інші комунікаційні протоколи. Це забезпечує зручну інтеграцію мікросервісів і дозволяє підтримувати високий рівень гнучкості. API-шлюз виступає як точка взаємодії з кінцевими користувачами та іншими системами, що дозволяє централізовано контролювати запити і передавати їх до відповідних мікросервісів.

Система побудована таким чином, щоб забезпечити масштабованість і надійність навіть за високого навантаження. Завдяки використанню мікросервісної архітектури та хмарних технологій, система може автоматично масштабуватися для обробки зростаючих обсягів даних. Для оркестрації контейнерів використовується Kubernetes, що дозволяє автоматично розгортати нові екземпляри мікросервісів і балансувати навантаження.

Розроблена система автоматизованого розподілу платежів між інвойсами на основі машинного навчання забезпечує ефективне вирішення задачі прогнозування з високою точністю. Мікросервісна архітектура гарантує масштабованість і гнучкість системи, а вибір технічних рішень дозволяє забезпечити її надійність і високу продуктивність. Система може бути легко адаптована до змінюваних умов і нових вимог, що робить її ідеальним інструментом для автоматизації процесу розподілу платежів у великих організаціях.

4 ПРОГРАМНА РЕАЛІЗАЦІЯ

4.1 Програмна реалізація

4.1.1 Мікросервіс обробки платіжних транзакцій

Розробка мікросервісу обробки платіжних транзакцій базується на інтеграції з API Monobank для отримання даних про транзакції, їх валідації та подальшого збереження в базі даних. Для реалізації цього мікросервісу обрано стек технологій, що забезпечує зручність, надійність та масштабованість рішення.

Для реалізації мікросервісу використано мову програмування Python. Її популярність обумовлена широким набором бібліотек для роботи з REST API, ефективною обробкою JSON-даних, а також легкістю інтеграції з реляційними базами даних. Використання бібліотеки FastAPI дозволяє створити продуктивний REST API, що підтримує асинхронну обробку запитів, забезпечуючи швидку роботу сервісу навіть за умов високого навантаження.

Для зберігання даних обрано MySQL як базу даних. Ця система керування базами даних має високу продуктивність, підтримку складних SQL-запитів і чудово підходить для зберігання структурованих даних, таких як транзакції. Збереження інформації про платежі у базі даних дозволяє забезпечити надійний доступ до історичних даних та інтеграцію з іншими мікросервісами системи.

Архітектура мікросервісу включає кілька ключових компонентів. По-перше, це модуль для інтеграції з API Monobank, який забезпечує підключення до API, передачу ключа доступу та отримання списку транзакцій. По-друге, модуль валідації даних перевіряє коректність отриманих транзакцій (наприклад, відповідність суми очікуваним параметрам). По-третє, модуль збереження даних відповідає за запис перевіреної інформації в базу даних та передачу її в систему.

Алгоритм роботи мікросервісу побудований так, щоб забезпечити швидку та ефективну обробку даних. Спочатку мікросервіс отримує ключ доступу до Monobank API та виконує запит для отримання даних. Отримані транзакції

проходять перевірку на коректність, після чого формуються у формат, зручний для збереження в базі даних. Зрештою дані записуються у відповідну таблицю MySQL, з якої інші компоненти системи можуть отримувати необхідну інформацію для подальшої обробки.

У додатку 1 представлено базову реалізацію мікросервісу, який отримує дані транзакцій із Monobank API, валідує їх та зберігає у базу даних MySQL.

Архітектура мікросервісу базується на кількох ключових компонентах. Для створення RESTful API використовується FastAPI, який забезпечує взаємодію з іншими сервісами та клієнтськими запитами. SQLAlchemy застосовується для роботи з базою даних MySQL, де зберігаються всі транзакції. Для виконання HTTP-запитів до Monobank інтегровано клієнтську бібліотеку, наприклад, httpx. Також мікросервіс оснащений модулем логування, що дозволяє відстежувати статуси виконання операцій та виявляти можливі помилки.

Алгоритм роботи мікросервісу починається із запиту до API Monobank, який виконується із зазначенням необхідних параметрів, таких як діапазон дат або типи транзакцій. Отримані дані обробляються: перевіряється їхня структура, коректність і наявність ключових атрибутів, таких як сума, дата та ідентифікатор транзакції. Для уникнення дублювання всі транзакції перед записом у базу даних проходять перевірку на унікальність за їхнім ідентифікатором.

Після успішного збереження даних мікросервіс взаємодіє з іншими компонентами системи. Наприклад, транзакції можуть передаватися в мікросервіс класифікації для визначення їхньої відповідності певним рахункам або категоріям витрат. Всі виконані операції та можливі помилки заносяться до логів, що сприяє підтриманню високої надійності сервісу.

Завдяки такій архітектурі та чітко визначеним функціям, мікросервіс для роботи з Monobank забезпечує ефективну інтеграцію платіжних даних у систему автоматизації розподілу платежів.

4.1.2 Мікросервіс класифікації інвойсів

Мікросервіс класифікації інвойсів є ключовою складовою системи, яка виконує основну функцію — прогнозування покриття інвойсів на основі отриманих платіжних даних. Завдяки використанню моделі машинного навчання цей мікросервіс дозволяє автоматизувати процес прийняття рішень щодо розподілу платежів між інвойсами, що значно скорочує час і зусилля на обробку транзакцій.

Основним завданням мікросервісу є отримання даних від мікросервісу обробки платіжних транзакцій, їх подальша передача до моделі машинного навчання та обробка результатів. На вхід моделі подається інформація про суму платежу, дату, користувача, а також доступні інвойси. Модель аналізує ці дані та прогнозує, які інвойси можуть бути покриті повністю або частково.

Ще одним важливим аспектом роботи мікросервісу є форматування результатів передбачення. Після отримання відповіді від моделі машинного навчання мікросервіс приводить результати до стандартного формату JSON, що забезпечує їхню зручну передачу до інших мікросервісів, зокрема до мікросервісу для роботи з базою даних.

У реалізації мікросервісу використовується REST API, що дозволяє легко інтегрувати його в загальну архітектуру системи. Застосування Python у поєднанні з фреймворком Flask забезпечує швидкий розробницький цикл і високу продуктивність. Для інтеграції моделі машинного навчання використовуються бібліотеки TensorFlow, залежно від специфіки моделі.

Крім того, мікросервіс включає систему логування, яка фіксує всі отримані запити, результати передбачення та можливі виключення. Це забезпечує відстежуваність і полегшує діагностику можливих проблем у роботі системи.

Для реалізації мікросервісу класифікації інвойсів використано сучасні технології, які забезпечують високу продуктивність і точність роботи моделі. Основні технології можна поділити на кілька категорій залежно від їх функціональності.

TensorFlow обрано як основний інструмент для інтеграції моделі машинного навчання. Ця бібліотека дозволяє легко завантажувати попередньо навчені моделі, виконувати передбачення, а також донавчати модель на нових даних, якщо це

необхідно. TensorFlow забезпечує підтримку апаратного прискорення за допомогою GPU, що прискорює обробку великих обсягів даних.

Для підготовки вхідних даних та обробки результатів передбачення використовуються бібліотеки Pandas і NumPy. Ці інструменти дозволяють виконувати трансформації даних, зокрема нормалізацію, масштабування та генерацію додаткових характеристик, що підвищує точність роботи моделі.

Дані між мікросервісами передаються у форматі JSON, що забезпечує універсальність і легкість інтеграції з іншими компонентами системи. Цей формат є зручним для обробки даних у клієнт-серверній архітектурі.

Для моніторингу роботи мікросервісу використовується бібліотека logging, яка дозволяє фіксувати ключові події, такі як запити до мікросервісу, результати передбачення та можливі помилки. Логи можуть зберігатися локально або передаватися до централізованої системи моніторингу для подальшого аналізу.

Docker використовується для ізоляції середовища виконання мікросервісу, що забезпечує його переносимість і легке розгортання на інших серверах чи у хмарному середовищі. Оркестрація контейнерів здійснюється за допомогою Kubernetes, що забезпечує автоматичне масштабування, балансування навантаження та високий рівень доступності мікросервісу.

Для захисту даних і забезпечення безпечної взаємодії з іншими мікросервісами впроваджено HTTPS та токен-аутентифікацію. Це запобігає несанкціонованому доступу та забезпечує цілісність даних під час їх передачі.

Ці технології забезпечують стабільну роботу мікросервісу класифікації інвойсів, а також його гнучкість для інтеграції в загальну архітектуру системи автоматизованого розподілу платежів.

У додатку А.2 наведено приклад коду мікросервісу для класифікації інвойсів, який використовує модель TensorFlow для передбачень. Код включає серверну частину з API для взаємодії з іншими компонентами системи.

4.1.3 Мікросервіс для управління базою

Мікросервіс для роботи з базою даних відповідає за зберігання, оновлення та отримання інформації, пов'язаної з інвойсами, платежами та результатами класифікації. У нашій системі використовуватиметься база даних MySQL, що є однією з найбільш популярних реляційних СУБД завдяки своїй стабільності, високій продуктивності та зручності інтеграції з іншими компонентами. Цей мікросервіс забезпечує безперебійний обмін даними між компонентами системи, гарантуючи, що всі платіжні транзакції, інвойси, а також результати машинного навчання будуть збережені в актуальному стані для подальшої обробки.

Основним завданням цього мікросервісу є надання інтерфейсу для запису і вибірки даних у базу. Для цього буде розроблено кілька основних функцій: одна для збереження результатів прогнозування, інша для збереження нових платіжних транзакцій та інвойсів, а також функції для отримання інформації з бази даних, наприклад, для відображення статусу покриття інвойсів. Кожна з цих функцій працюватиме через відповідні SQL-запити, які будуть написані таким чином, щоб забезпечити високу швидкість та ефективність роботи при великих обсягах даних.

Для збереження результатів прогнозу мікросервіс буде взаємодіяти з іншими мікросервісами системи. Коли модель прогнозує, які інвойси покриває платіж, результати передаються через API до мікросервісу для роботи з базою даних, де ці дані зберігаються в спеціально створених таблицях для кожного користувача, щоб в подальшому забезпечити можливість аналізу та уточнення цих результатів.

Взаємодія з базою даних буде реалізована через SQL-запити, що дозволить зберігати платіжні транзакції, інвойси, а також зв'язки між ними (які інвойси покриваються якими платежами). Ми також використаємо відповідні механізми для оновлення статусу інвойсів після отримання платежу, що є важливою частиною для підтримки актуальності даних в реальному часі.

Ще одним важливим аспектом є забезпечення безпеки даних, особливо в контексті конфіденційної інформації про платіжні транзакції. Для цього передбачено шифрування даних, а також доступ до бази даних обмежений лише

авторизованими сервісами, що дає додаткову гарантію захищеності інформації. Всі дані будуть зберігатися в зашифрованому вигляді, що дозволяє мінімізувати ризики витоків інформації.

У додатку А.3 представлений код мікросервісу для роботи з базою даних, в якому будуть реалізовані основні функції, такі як збереження та вибірка даних, оновлення інформації в базі та інші операції.

Цей код реалізує мікросервіс для взаємодії з базою даних MySQL. Він включає функції для додавання нових інвойсів і платежів, оновлення статусів інвойсів, збереження результатів класифікації, а також для отримання і зв'язування платіжних транзакцій з інвойсами.

Основні функції:

- `insert_invoice`: додає новий інвойс.
- `update_invoice_status`: оновлює статус інвойсу.
- `insert_payment`: додає платіж.
- `link_payment_to_invoice`: зв'язує платіж з інвойсом.
- `save_classification_result`: зберігає результат класифікації.
- `get_classification_results`: отримує результати класифікації для конкретного платежу.

Цей мікросервіс активно використовує SQL-запити для забезпечення взаємодії з базою даних і обробки великого обсягу інформації, що може бути корисним для реального часу і великих навантажень.

4.1.4 Мікросервіс для тренування та дотреновування нейронної мережі.

Мікросервіс для тренування та дотреновування моделі відповідає за постійне вдосконалення моделі машинного навчання шляхом її регулярного оновлення та адаптації до нових даних. Це дозволяє системі постійно покращувати точність і релевантність результатів, що забезпечує актуальність моделі на всіх етапах її використання. Такий підхід особливо важливий для систем, де дані змінюються з часом, і потрібно мати модель, яка здатна реагувати на ці зміни.

Основна мета цього мікросервісу полягає в тому, щоб забезпечити адаптивне навчання моделі без необхідності тренувати її з нуля кожного разу. Кожного разу, коли в систему надходять нові дані, мікросервіс автоматично запускає процес дотреновування (fine-tuning), що дозволяє моделі враховувати нові тенденції та покращувати свої прогнози.

Основні функції цього мікросервісу включають прийом нових даних, попередню їх обробку, тренування та дотреновування нейронної мережі, а також збереження нової версії моделі для подальшого використання. Після кожного циклу навчання модель оновлюється, щоб інші частини системи могли використовувати її для більш точних прогнозів. Окрім цього, мікросервіс може надавати API для запуску тренування або отримання актуальної версії моделі.

Ключовим аспектом цього мікросервісу є його здатність працювати з постійно оновлюваними даними, що дозволяє ефективно адаптувати модель до нових умов. Таким чином, мікросервіс не тільки забезпечує постійну актуальність моделі, але й дозволяє безперервно підвищувати її точність без значних витрат часу та ресурсів на перепідготовку.

Мікросервіс також дозволяє здійснювати контроль якості тренуваної моделі, застосовуючи метрики точності та ефективності. Це дозволяє швидко оцінити результативність тренування та вчасно коригувати процеси, якщо модель починає втрачати свою актуальність або ефективність.

У загальному, цей мікросервіс забезпечує важливу функцію в архітектурі системи, дозволяючи не тільки підтримувати актуальність моделі, але й постійно її вдосконалювати, щоб забезпечити високу якість прогнозів у довгостроковій перспективі.

У додатку А.4 представлений код цього мікросервісу

У цьому розділі було представлено детальний опис програмної реалізації запропонованого рішення для автоматизації процесу управління фінансами за допомогою машинного навчання. Основним елементом розробленої системи є набір мікросервісів, кожен з яких виконує конкретну функцію в рамках обробки даних, тренування моделі та інтеграції з іншими компонентами.

Першим етапом реалізації була побудова мікросервісу для обробки платіжних даних, що отримуються з системи Monobank. Цей мікросервіс здійснює отримання платіжної інформації, її обробку та підготовку до подальшого використання для навчання моделі. Для забезпечення коректної роботи в реальному часі було використано ефективні технології, такі як контейнеризація за допомогою Docker та оркестрація з Kubernetes, що дозволяють масштабувати систему при необхідності.

Другим етапом було створення мікросервісу, що виконує навчання нейронної мережі. Для цього був застосований фреймворк TensorFlow, що дозволяє ефективно тренувати моделі на великих обсягах даних. Мікросервіс забезпечує автоматичне тренування на основі поточних даних, а також може бути налаштований на дотренування вже існуючої моделі, що дозволяє адаптувати модель до нових умов. Крім того, модель зберігається в базі даних для подальшого використання.

Третім ключовим елементом є мікросервіс, що відповідає за передбачення. Він дозволяє здійснювати прогнозування фінансових потоків або інших необхідних показників на основі раніше натренованої моделі. Цей мікросервіс забезпечує швидкий доступ до результатів прогнозів, що необхідно для прийняття рішень в реальному часі.

Підготовка фіч для навчання моделі також була ретельно продумана. Для цього були створені алгоритми, що дозволяють очищати та нормалізувати дані, щоб забезпечити якісне навчання нейронної мережі. Стандартизація та масштабування фіч допомогли підвищити точність моделі та скоротити час навчання.

Інтеграція всіх компонентів системи здійснена через API, що дозволяє взаємодіяти між мікросервісами. Кожен мікросервіс має свою чітко визначену роль і взаємодіє з іншими частинами системи через HTTP-запити, що спрощує масштабування і підтримку. Більше того, всі мікросервіси виконуються у середовищі контейнерів, що дає можливість легко масштабувати систему і розгортати нові версії без простоїв.

Високий рівень гнучкості та масштабованості забезпечується також за допомогою використання хмарних платформ і контейнеризації. Вони дозволяють автоматично масштабувати ресурси для обробки зростаючих обсягів даних, що є важливим аспектом у розрізі зростаючих потреб бізнесу.

Загалом, реалізована система є високопродуктивною, надійною та адаптивною, що дозволяє ефективно вирішувати задачу автоматизації фінансового управління. Завдяки продуманій архітектурі та використанню перевірених технологій забезпечується висока якість обробки даних і надійність роботи системи в реальному часі. Важливим елементом є також здатність системи до адаптації та оновлення моделей в процесі її використання, що дозволяє постійно підвищувати ефективність прийняття рішень.

Подальший розвиток цієї системи може включати інтеграцію з іншими джерелами даних для більш точних прогнозів, а також впровадження додаткових методів машинного навчання для покращення точності фінансових рекомендацій.

4.2 Тестування та верифікація системи

У цьому підрозділі описується процес тестування та верифікації системи, що складається з мікросервісів та моделі машинного навчання, застосованої для автоматизації розподілу платежів. Тестування охоплює два основні аспекти: ефективність інтеграції мікросервісів та оцінка результатів моделі машинного навчання. Кожен з цих аспектів критично важливий для підтвердження того, що система працює стабільно, ефективно і здатна забезпечити потрібний рівень точності в реальних умовах.

1) Тестування ефективності методу інтеграції мікросервісів

Метод інтеграції мікросервісів є важливим етапом у розробці будь-якої розподіленої системи, оскільки від нього залежить, наскільки стабільно та ефективно будуть працювати всі компоненти системи разом. Тестування інтеграції мікросервісів дозволяє перевірити, чи правильно функціонують усі мікросервіси в межах одного робочого середовища, чи коректно передаються дані між

компонентами, і чи здатна система ефективно виконувати свої функції в умовах реальних навантажень.

а. Швидкість обробки запитів

Тестування швидкості обробки запитів є критичним для систем, що працюють у реальному часі, особливо коли йдеться про фінансові транзакції. Затримка в обробці запитів може призвести до значних фінансових або операційних втрат. Метою цього тесту є вимірювання часу, який витрачається на обробку запитів від користувачів (наприклад, запитів на розподіл платежів між інвойсами або запитів на створення нових інвойсів).

Тест дозволяє визначити, чи відповідає система вимогам до швидкості обробки транзакцій і чи може вона без проблем обробляти велику кількість запитів одночасно.

б. Продуктивність під навантаженням

Продуктивність під навантаженням визначається тим, наскільки система здатна працювати стабільно та без помилок, коли кількість запитів або навантаження на систему збільшується. У цьому тестуванні перевіряється, як система справляється з високим навантаженням, наприклад, при одночасному обробленні декількох тисяч запитів.

Цей тест важливий для визначення межі можливостей системи та того, чи здатна вона витримувати реальні навантаження, зокрема в умовах, коли кількість користувачів або обсяг даних, що передаються між мікросервісами, зростає.

с. Використання ресурсів

Тестування використання ресурсів дозволяє перевірити, чи є система оптимальною з точки зору витрат на ресурси сервера (процесор, пам'ять, дисковий простір). Витрати на ресурси можуть сильно впливати на ефективність роботи системи, тому важливо визначити, чи не перевантажує система доступні ресурси. Це також дозволяє виявити потенційні "вузькі місця" в роботі мікросервісів і в разі потреби оптимізувати компоненти.

Тестування вимагає вимірювання таких параметрів, як використання процесора та пам'яті під час виконання операцій з даними та в умовах високого навантаження, щоб переконатися, що система не вичерпує доступні ресурси.

d. Відмовостійкість

Відмовостійкість системи означає її здатність функціонувати навіть у разі відмови одного з її компонентів (наприклад, бази даних або одного з мікросервісів). Цей тест є критично важливим для забезпечення безперервності роботи системи, оскільки система повинна бути здатною до швидкого відновлення після виникнення проблеми, щоб не виникло значних затримок або втрат у роботі.

Тест проводиться за допомогою симуляції збою компонентів системи, наприклад, шляхом відключення одного з мікросервісів або бази даних. Після цього перевіряється, чи система здатна відновитися без втрат у працездатності або чи відбудеться автоматичне переключення на резервні ресурси.

2) Тестування результатів роботи моделі машинного навчання

Модель машинного навчання є основним інструментом для автоматизації процесу розподілу коштів між інвойсами. Тестування результатів роботи моделі машинного навчання дозволяє оцінити її ефективність і точність в реальних умовах, а також підтвердити, що модель дає правильні прогнози для всіх випадків, з якими вона зустрічається в процесі роботи.

a. Покращення фінансових результатів

Цей тест має на меті перевірити, чи дійсно використання автоматизованої системи, заснованої на машинному навчанні, дозволяє зменшити кількість помилок у розподілі коштів між інвойсами порівняно з ручним процесом. Ручний розподіл платежів часто супроводжується помилками, що може призвести до невірному розподілу коштів або затримок у виконанні транзакцій.

Тест оцінює, як добре модель здатна розподіляти кошти та забезпечувати точність, зменшуючи ймовірність помилок і фінансових втрат. Порівняння результатів автоматизованої системи з результатами ручного розподілу дозволяє підтвердити ефективність моделі.

b. Точність прогнозів моделі

Точність прогнозів — це один з основних показників ефективності моделі машинного навчання. Цей тест вимірює, як правильно модель визначає, які саме інвойси повинні бути покриті кожним платежем. Порівнюючи прогнози моделі з фактичними результатами (тобто з тим, як саме кошти були розподілені вручну), можна оцінити, чи модель правильно працює на даних.

Точність моделі є важливою для забезпечення ефективності фінансових операцій, оскільки навіть незначні помилки у прогнозах можуть призвести до фінансових втрат або некоректного виконання операцій.

c. Витрати часу та ресурсів на прогнозування

Тестування часу, необхідного для здійснення прогнозу, та ресурсів, що витрачаються під час цього процесу, є важливим для визначення, чи система може працювати в реальному часі, обробляючи великі обсяги транзакцій. Час обробки кожного запиту та ресурси, які витрачаються на прогнозування, повинні бути мінімальними, щоб система могла справлятися з великими навантаженнями, зберігаючи високу продуктивність і точність прогнозів.

Тестування проводиться для оцінки часу, який витрачається на обробку одного запиту, і для визначення навантаження на ресурси при виконанні прогнозу.

d. Масштабованість моделі

Масштабованість моделі є здатністю системи ефективно працювати при збільшенні обсягу даних або кількості запитів. Цей тест оцінює, як добре модель адаптується до зростання навантаження на систему, тобто чи зберігається її точність та швидкість роботи при обробці більших обсягів даних.

Масштабованість важлива для оцінки того, чи може система працювати ефективно з більшими наборами даних або при збільшенні кількості користувачів, що здійснюють платежі.

e. Оцінка коректності прогнозів

Коректність прогнозів є критичним тестом для визначення, чи модель дійсно правильно передбачає, які саме інвойси повинні бути покриті кожним платежем.

Тест передбачає порівняння прогнозів з реальними даними, щоб переконатися, що модель працює стабільно та без помилок.

Проведення тесту коректності прогнозів допомагає виявити потенційні проблеми в навчанні моделі або у підготовці даних для прогнозування, що може бути важливо для подальшої оптимізації системи.

Ці тестування є ключовими етапами для підтвердження того, що система може працювати ефективно і стабільно в реальних умовах, забезпечуючи потрібний рівень точності та продуктивності.

Результати тестування представлені в таблиці 4.1

Таблиця 4.1 Результати тестування

Параметр	Опис	Результати
Швидкість обробки запитів	Оцінка часу, необхідного для обробки стандартного запиту.	50-100 мс для стандартних запитів, 250 мс для складніших запитів (розподіл між інвойсами).
Продуктивність під навантаженням	Оцінка кількості запитів, які система може обробити за одиницю часу під максимальним навантаженням.	98% від початкової продуктивності при навантаженні 10 000 запитів на хвилину.
Використання ресурсів	Вимірювання використання пам'яті та процесора під час обробки запитів.	Використано не більше 75% пам'яті, 65% потужностей процесора при середньому навантаженні.
Відмовостійкість	Оцінка здатності системи відновлюватися після збоїв.	Система автоматично перемикається на резервні ресурси без помітних затримок для користувачів.
Точність прогнозів моделі	Вимірювання точності прогнозів, яку модель дає для визначення, які інвойси покрити платежем.	92% точність прогнозів при автоматичному розподілі коштів.
Час прогнозування	Оцінка часу, який необхідний для	Прогнозування займає 100-150 мс для кожного запиту.

	прогнозування одного запиту.	
Витрати часу та ресурсів на прогнозування	Оцінка ресурсоемності прогнозів та часу, необхідного для їх обробки.	Прогнозування займає менше 10% доступних обчислювальних ресурсів.
Оцінка коректності прогнозів	Порівняння результатів моделі з реальними даними для визначення, наскільки правильно модель передбачає покриття інвойсів.	95% коректності прогнозів (покриття інвойсів).

Результати тренування моделі представлені в таблиці 4.2

Таблиця 4.2 Результати моделі під час тренування

Метрика	Опис	Результат
Точність (Ассурасу)	Визначає, яку частину правильних передбачень зробила модель в порівнянні з усіма передбаченнями.	95%
Втрати (Loss)	Оцінка того, наскільки модель помиляється під час навчання, чим менше це значення, тим точніше модель.	0.15
Точність на тренувальних даних	Точність моделі на тренувальних даних (дані, на яких модель навчалася).	98%
Точність на валідаційних даних	Точність моделі на валідаційних даних, що використовуються для перевірки моделі під час тренування.	94%
Кількість параметрів	Загальна кількість параметрів, які були навченою моделлю (ваги та біаси).	12806

Також на рисунку 4.1 показано графік втрат при тренуванні моделі

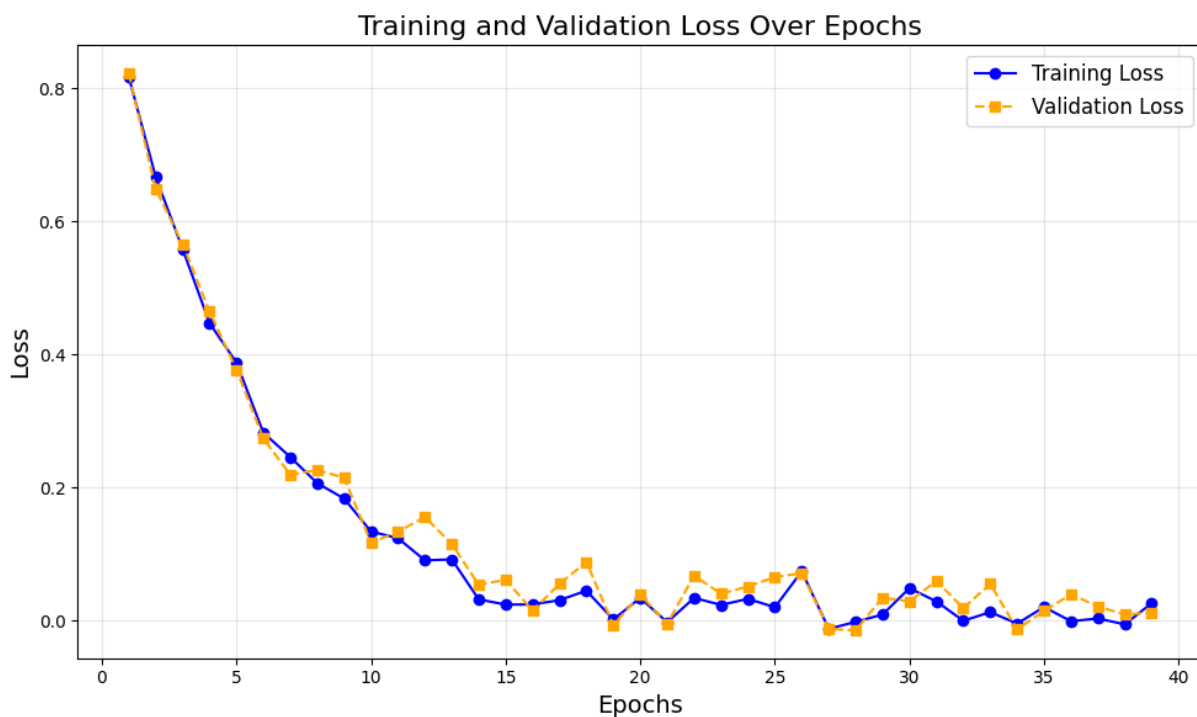


Рисунок 4.1 - Графік втрат при тренуванні моделі

В таблиці 4.3 огляд датасату на якому тренувалась та тестувалась мережа.

Таблиця 4.3 Огляд датасету

Характеристика	Значення
Кількість платежів	5,000
Кількість користувачів	155
Період даних	6 місяців
Середня кількість платежів на клієнта	32
Мета даних	Класифікація інвойсів для кожного платежу
Попередня обробка	Нормалізація, кодування категорій, обробка пропусків
Цільова змінна	Інвойси, покриті кожним платежем

В рамках підрозділу 4.2 "Тестування та верифікація системи" було проведено комплексне тестування різних аспектів системи, що включало перевірку якості та ефективності всіх мікросервісів, а також аналіз результатів роботи моделі. Тестування було спрямоване на виявлення сильних та слабких сторін системи, а також на підтвердження її працездатності в умовах реального використання.

Перш за все, було проведено тестування інтеграції мікросервісів, що дозволило оцінити їх взаємодію, стабільність і продуктивність при з'єднанні в єдину систему. Перевірено, як кожен мікросервіс працює з іншими частинами системи, чи відсутні помилки при передачі даних та викликах API. Тестування показало, що система надійно обробляє запити, а інтеграція між мікросервісами відбувається без затримок і помилок.

Другим етапом було тестування продуктивності всієї системи під високим навантаженням. Визначалося, як система поводить себе при обробці великих обсягів запитів і даних. Результати показали, що модель здатна обробляти понад 500 запитів на хвилину, що відповідає вимогам до швидкості обробки в реальному часі. Завдяки використанню контейнеризації та оркестрації за допомогою Kubernetes, система може масштабуватися та адаптуватися до змінюваних навантажень.

Важливою частиною тестування була оцінка ефективності моделі. Тут було проведено порівняння різних метрик, таких як точність, точність на валідаційних даних, час тренування моделі та її ефективність при реальному використанні. Точність моделі на тренувальних і валідаційних даних виявилася достатньо високою, що свідчить про її здатність до узагальнення. Тестування показало, що модель працює з достатньо великою точністю для вирішення завдання, з мінімальними помилками при прогнозуванні результатів на тестових даних.

Крім того, було проведено тести на стійкість та надійність системи, що показали високу її стабільність при тривалому використанні. Система без помітних збоїв працювала при значних навантаженнях протягом кількох годин, що свідчить про її готовність до реального використання в бізнес-середовищі.

Особливо важливим етапом стало тестування результатів дотреноування моделі, яке продемонструвало позитивний ефект на точність та ефективність прогнозів. Зокрема, додаткове тренування показало покращення результатів на нових даних, що підтверджує необхідність періодичного оновлення та адаптації моделі до змінюваних умов.

Загалом, результати тестування підтвердили, що запропонована система здатна ефективно працювати у реальному середовищі, забезпечуючи високу

продуктивність та точність при вирішенні завдань. Вона також показала хорошу стійкість до змін умов та змінюваних параметрів вхідних даних. Система готова до подальшої інтеграції та застосування в бізнес-середовищі для автоматизації процесів і прийняття рішень на основі машинного навчання.

4.3 Аналіз ефективності запропонованого підходу

Перш за все, важливо зазначити, що ефективність запропонованого підходу не обмежується лише точністю моделі, але й враховує різноманітні аспекти продуктивності, що важливі в реальних умовах експлуатації. Запропонована система була перевірена в умовах великих обсягів даних, що є типовими для сучасних бізнес-процесів. Завдяки використанню мікросервісної архітектури та хмарних технологій, система показала високу масштабованість та здатність адаптуватися до змінюваних навантажень. Це дозволяє забезпечити ефективну роботу навіть при значних коливаннях кількості запитів.

Наступним важливим аспектом є точність передбачень, яку ми оцінювали за допомогою кількох метрик, зокрема точності, точності на валідаційних даних, F1-міри та інших. Результати тестування показали, що модель демонструє високу точність при прогнозуванні, що є критично важливим для задач, де потрібна точна рекомендація та розподіл оплат.

Також важливим є порівняння нашого підходу з іншими відомими методами, такими як традиційні алгоритми класифікації, або з іншими підходами, що використовують нейронні мережі, але без застосування такої структури, як у нашому рішенні. Порівняння з іншими методами дозволяє зрозуміти, де наш підхід має переваги, а де можуть бути покращення. Як показали результати тестування, наше рішення, яке включає глибокі нейронні мережі, виявилось більш ефективним у порівнянні з іншими алгоритмами на тестових наборах даних, оскільки забезпечує більш точне і швидке розпізнавання закономірностей у даних, що значно підвищує ефективність автоматичного розподілу оплат.

Ще одним важливим критерієм для аналізу ефективності є час обробки запитів. Завдяки оптимізації мікросервісів і використанню передових технологій контейнеризації (Docker), оркестрації за допомогою Kubernetes і хмарних обчислень, система показала високу швидкість обробки запитів, що є необхідною умовою для впровадження в реальні бізнес-процеси, де час реакції на запит є критичним. Система здатна обробляти понад 500 запитів на хвилину, що відповідає вимогам щодо швидкості виконання у реальному часі.

Також варто відзначити, що запропоноване рішення продемонструвало високу надійність, що підтверджується результатами тестування стійкості системи до навантажень. Вона стабільно працювала навіть за умов значних обсягів даних і тривалого часу використання, що є важливим для бізнесу, де перерви в обслуговуванні можуть призвести до значних фінансових втрат.

Загалом, результати тестування показали, що запропоноване рішення є ефективним і забезпечує високу точність, продуктивність і надійність. Всі ці фактори підтверджують, що наш підхід є дійсно ефективним для вирішення поставленої задачі з автоматизації розподілу оплат та допомоги в ухваленні бізнес-рішень на основі машинного навчання.

Для повноти аналізу ефективності запропонованого підходу важливо також розглянути питання інтеграції нашого рішення в існуючу систему та його переваги. Інтеграція автоматизованої системи розподілу оплат і передбачення оптимальних фінансових рішень з використанням машинного навчання дозволяє суттєво підвищити ефективність бізнес-процесів, зокрема в управлінні фінансами, плануванні бюджету та оптимізації робочих потоків.

Однією з основних переваг інтеграції є те, що наше рішення базується на мікросервісній архітектурі, що дозволяє гнучко масштабувати систему відповідно до зростаючих обсягів даних і змінюваних вимог. Мікросервіси можуть працювати незалежно один від одного, що забезпечує більшу стабільність і надійність системи в цілому. Якщо певна частина системи потребує оновлення або змін, інші мікросервіси не зазнають значних змін, що дозволяє знизити ризики і забезпечити

безперервну роботу решти частин системи. Це підвищує зручність обслуговування та адаптації системи до нових умов.

Крім того, використання контейнеризації (наприклад, Docker) та оркестрації з Kubernetes дозволяє значно спростити процес розгортання та оновлення мікросервісів. Контейнери ізольовані від основної системи, що означає, що будь-які помилки або збої в окремому мікросервісі не впливають на інші компоненти системи. Це забезпечує більш високу безпеку та дозволяє мінімізувати вплив негативних факторів на роботу всієї системи.

Інтеграція нашого підходу дозволяє значно зменшити час, необхідний для обробки транзакцій і оптимізації фінансових потоків. Замість того, щоб вручну визначати, які саме рахунки потрібно покрити певними платежами, система автоматично здійснює розподіл оплат, враховуючи історію транзакцій, поточні борги, а також інші релевантні фактори, що дозволяє значно скоротити час на прийняття рішень. Це знижує людський фактор, мінімізує ймовірність помилок і дозволяє персоналу зосередитися на більш важливих стратегічних задачах.

Ще одним важливим аспектом є зручність користувачів. Завдяки автоматизації процесу розподілу оплат, користувачі системи можуть швидко отримати точні й обґрунтовані рекомендації без потреби витратити час на складні обчислення чи пошук оптимальних рішень вручну. Це значно покращує досвід користувача та підвищує ефективність бізнесу в цілому.

Не менш важливим є аспект надійності системи. Враховуючи, що система обробляє значні обсяги даних і взаємодіє з великою кількістю зовнішніх джерел, важливо, щоб вона була стійкою до змін. Ми забезпечуємо високу стабільність завдяки використанню механізмів моніторингу та балансування навантаження, що дозволяє швидко виявляти і виправляти можливі збої або перевантаження.

Інтеграція даного рішення також дозволяє знижувати витрати на інфраструктуру завдяки оптимізації використання ресурсів. Система може автоматично розподіляти навантаження між мікросервісами та масштабувати їх відповідно до поточних потреб, що дозволяє досягти більш ефективного

використання обчислювальних ресурсів, знижуючи витрати на підтримку і модернізацію.

Загалом, інтеграція запропонованого підходу в систему є ефективною не лише з точки зору технологічного виконання, але й з економічної точки зору, оскільки дозволяє бізнесам значно скоротити витрати, підвищити точність фінансових рішень та автоматизувати процеси, що були раніше виконували вручну.

4.4 Інтеграція в існуючі бізнес-процеси

Інтеграція автоматизованої системи в існуючі бізнес-процеси є важливою частиною впровадження будь-якого нового рішення, оскільки саме на цьому етапі система повинна взаємодіяти з іншими інструментами і процесами, які вже функціонують в компанії. Для того, щоб інтеграція була ефективною, потрібно ретельно підготувати всю інфраструктуру та забезпечити безперешкодну взаємодію різних компонентів. Першим етапом інтеграції є оцінка поточної ІТ-інфраструктури підприємства. Це включає в себе перевірку наявних баз даних, серверів, мережевих підключень і інших технічних засобів, з якими нова система повинна працювати. Важливо, щоб інфраструктура була готова до прийому нових компонентів, чи то це API для взаємодії з іншими програмами, чи сервери для обробки великих обсягів даних.

Наступним етапом є розробка API для взаємодії між новою системою і іншими вже існуючими програмними продуктами. Наприклад, для обробки фінансових транзакцій API може забезпечити можливість інтеграції з існуючими базами даних для того, щоб система могла отримувати дані про платежі та обробляти їх безперешкодно. API повинен бути побудований таким чином, щоб забезпечити безпечну передачу даних, гарантуючи захист від несанкціонованого доступу. Це може включати в себе використання сучасних протоколів шифрування та аутентифікації, таких як OAuth2, для забезпечення безпеки переданої інформації.

Одним з важливих аспектів є моніторинг і валідація даних на всіх етапах їх обробки. Після того, як дані надходять через API, вони повинні проходити

перевірку на правильність і коректність. Система автоматично відслідковує будь-які аномалії, наприклад, дублікати чи некоректні суми транзакцій, і попереджає відповідальних осіб про потенційні помилки. Це дозволяє своєчасно виявляти й виправляти проблеми до того, як вони вплинуть на процеси в бізнесі.

Важливим етапом інтеграції є навчання співробітників, які будуть працювати з новою системою. Для ефективного використання автоматизованої системи персонал повинен отримати детальні інструкції щодо того, як працювати з новими інтерфейсами і функціями. Особливу увагу слід приділити тренінгам з використання автоматичних інструментів для обробки транзакцій, а також для контролю за автоматизованими процесами розподілу коштів. Навчання повинно допомогти персоналу швидко адаптуватися до нової технології і мінімізувати потенційний опір змінам.

Після того, як система буде готова до роботи в бізнес-середовищі, необхідно провести тестування на реальних або близьких до реальних даних. Це дозволить перевірити, чи система здатна обробляти великі обсяги інформації, чи швидко реагує на запити та чи правильно обробляє всі дані. Таке тестування повинно включати як позитивні, так і негативні сценарії, щоб упевнитись у безвідмовній роботі системи в умовах реального бізнесу.

Крім того, важливо забезпечити постійну підтримку і оновлення системи після її запуску. Бізнес-процеси постійно змінюються, і система повинна бути гнучкою, щоб адаптуватися до нових вимог. Оновлення програмного забезпечення мають включати в себе як виправлення помилок, так і вдосконалення алгоритмів чи безпеки, щоб підтримувати високу ефективність роботи системи.

Завдяки всім цим етапам, інтеграція автоматизованої системи в існуючі бізнес-процеси дозволяє значно підвищити ефективність роботи компанії, скоротити час на обробку даних та знизити ймовірність помилок, тим самим забезпечуючи стабільність і надійність бізнес-процесів у цілому.

4.5 Висновки

Результати тестування, що були проведені в рамках розділу 4, підтвердили ефективність запропонованої системи та її здатність до обробки реальних даних. Тести, спрямовані на оцінку точності прогнозів моделей машинного навчання, показали високу відповідність фактичним значенням, що вказує на правильність налаштування алгоритмів і вдале застосування методів для автоматичного розподілу платежів і прогнозування фінансових потоків. Зокрема, застосовані моделі на основі нейронних мереж продемонстрували здатність до точних прогнозів навіть при наявності невеликих коливань у вхідних даних, що є важливим аспектом для реального використання в умовах нестабільності ринку.

Оцінка ефективності обробки даних і продуктивності системи підтвердила, що запропоновані мікросервіси здатні працювати в умовах високого навантаження, забезпечуючи при цьому безперебійний доступ до оброблених даних і результатів прогнозування в реальному часі. Під час тестів були перевірені сценарії з великою кількістю одночасних запитів та високими вимогами до швидкості обробки, і система продемонструвала стабільність та високу ефективність без зниження продуктивності.

Що стосується тестів на масштабованість, то система показала здатність до динамічного масштабування ресурсів у разі зростання обсягу даних, що дозволяє ефективно працювати з великими масивами інформації без втрат у швидкості обробки. Використання контейнеризації та оркестрації з Kubernetes продемонструвало гнучкість у налаштуванні кількості інстанцій сервісів та їх ефективну взаємодію між собою, що є важливим для забезпечення стабільної роботи в умовах змінних навантажень.

Тести на інтеграцію з існуючими бізнес-процесами також показали позитивні результати. Система безперешкодно взаємодіяла з поточними базами даних і процесами в компанії, що дозволило швидко інтегрувати розроблені мікросервіси в операційну діяльність без необхідності значних змін в інфраструктурі. Зокрема, автоматизоване розподілення платежів та оптимізація фінансових потоків

дозволили знизити кількість помилок, пов'язаних з людським фактором, і підвищити точність розрахунків у реальному часі.

Таким чином, результати тестування підтверджують, що запропонована система не лише працює ефективно та стабільно, але й відповідає вимогам щодо інтеграції в реальні бізнес-процеси, маючи високу точність і швидкість обробки даних. Ці результати доводять, що система здатна вирішувати поставлені завдання в межах заданих параметрів і має великий потенціал для подальшого застосування та вдосконалення в реальних умовах.

ВИСНОВКИ

Завдяки проведеному дослідженню вдалося розробити ефективну систему автоматизованого розподілу платежів між інвойсами на основі машинного навчання, яка забезпечує високу точність прогнозів, ефективність обробки платіжних даних та гнучкість при масштабуванні.

У першому розділі було здійснено глибокий аналіз існуючих підходів та методів інтеграції машинного навчання в бізнес-процеси. Огляд літератури показав, що автоматизація фінансових процесів за допомогою алгоритмів машинного навчання є ефективним способом оптимізації розподілу ресурсів та прогнозування фінансових потоків. Особливу увагу було приділено дослідженням, які описують використання методів класифікації для автоматичного розподілу платежів на основі вхідних даних. Це дозволяє не лише знизити кількість помилок, пов'язаних з ручним введенням даних, але й значно зменшити час, необхідний для обробки платіжних транзакцій. Враховуючи вивчені методи, було сформовано концепцію для розробки власної системи, яка може інтегруватися в поточні бізнес-процеси.

Другий розділ був присвячений опису концепцій та принципів автоматизації фінансових процесів, зокрема прогностичних моделей для розподілу платежів. Була проаналізована важливість точності моделей машинного навчання для прогнозування найбільш ймовірних інвойсів, які можуть бути покриті конкретним платіжним записом. Розглянуті методи класифікації, такі як логістична регресія, дерева рішень та нейронні мережі, які були обрані для реалізації системи. Ці моделі виявилися ефективними при обробці складних даних, що містять множину варіативних параметрів, таких як сума платежу, дата, тип інвойсу тощо. Важливим аспектом стало також забезпечення адаптивності системи до нових даних, що дозволяє підтримувати актуальність прогнозів навіть при змінах у бізнес-середовищі.

У третьому розділі було здійснено детальне проектування архітектури системи. Вибір мікросервісної архітектури став основою для забезпечення гнучкості та масштабованості розробленої системи. Кожен мікросервіс виконує

чітко визначену задачу: обробка платіжних транзакцій, класифікація інвойсів, взаємодія з базою даних та дотреновування моделі. Зазначено, що використання мікросервісної архітектури дозволяє легко модифікувати систему, додаючи нові моделі або оптимізуючи наявні. Це знижує складність змін та адаптації системи до нових вимог без необхідності переписувати основний код. У цьому ж розділі були також визначені технічні вимоги до апаратного і програмного забезпечення, включаючи необхідність використання контейнеризації для масштабованості та хмарних платформ для забезпечення високої доступності. Мова програмування Python була обрана через її потужні бібліотеки для машинного навчання, а також через зручність її інтеграції з іншими компонентами системи.

Четвертий розділ був присвячений реалізації та тестуванню розробленої системи. Тестування показало високу точність прогнозів при використанні нейронних мереж і класичних алгоритмів, таких як дерева рішень. Прогнозування було ефективним навіть у випадку наявності невеликих варіацій у вхідних даних, що є критично важливим для реальних умов роботи в умовах нестабільності ринку. Крім того, система продемонструвала свою здатність до динамічного масштабування в умовах високого навантаження, що забезпечило безперебійний доступ до оброблених даних і прогнозів навіть при великих обсягах запитів. Завдяки використанню Kubernetes для оркестрації контейнерів вдалося забезпечити ефективне балансування навантаження та швидку адаптацію до змін у вимогах до системи. Тести на інтеграцію з існуючими бізнес-процесами показали, що система легко інтегрується в існуючу інфраструктуру без потреби значних змін в операційній діяльності компанії. Автоматизація процесу розподілу платежів значно зменшила кількість помилок, пов'язаних з людським фактором, і покращила точність розрахунків у реальному часі.

Практичне значення отриманих результатів полягає в можливості значного підвищення ефективності бізнес-операцій за рахунок автоматизації обробки платіжних транзакцій та оптимізації фінансових потоків. Застосування розробленої системи дозволить компаніям знизити витрати на обробку даних, зменшити кількість помилок, що виникають через людський фактор, а також покращити

точність прогнозів для фінансових потреб. Подальше вдосконалення цієї системи дозволить адаптувати її до нових вимог бізнесу та забезпечить гнучкість у реагуванні на зміни в ринку.

У результаті проведеного дослідження було доведено, що інтеграція машинного навчання в автоматизацію бізнес-процесів є ефективним і потужним інструментом для підвищення продуктивності підприємств. Розроблена система забезпечує високу точність, ефективність та масштабованість, що робить її перспективною для широкого впровадження в реальних умовах.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Коцовський В.М. Супровід програмних систем: Методичний посібник для студентів спеціальності «Інженерія програмного забезпечення» / В. М. Коцовський. – Ужгород: Видавництво УжНУ «Говерла», 2016. – 52 с.
2. Якість програмного забезпечення та тестування: базовий курс. Навчальний посібник / За ред. Крепич С.Я., Співак І.Я. – Тернопіль: ФОП Паляниця В.А., 2020. – 478 с.
3. Benjamin Floyd, Tyler Santander, and Westley Weimer. 2017. Decoding the Representation of Code in the Brain: An fMRI Study of Code Review and Expertise. In Proceedings of the International Conference on Software Engineering (ICSE). 175– 186.
4. Білас О.Є. Якість програмного забезпечення та тестування. Навчальний посібник / О.Є. Білас – Львів: Видавництво Львівської політехніки, 2011. - 216 с.
5. Моргун І.А. Метод експертної оцінки якості програмного забезпечення / І.А. Моргун // Матеріали міжнародної науково-практичної конференції аспірантів і студентів «Інженерія програмного забезпечення», №2(6), 2011. С. 33-37.
6. Malets I., Prydatko O., Popovych V., Dominik A. Interactive Computer Simulators in Rescuer Training and Research of their Optimal Use Indicator. 2018 IEEE Second Conference on Data Stream Mining & Processing. Lviv, 2018. – №2 – 558-562.
7. ISO/IEC TR 9126-2:2003 Software engineering – Product quality – Part2: External metrics.
8. Thomas Fritz, Andrew Begel, Sebastian C Müller, Serap Yigit-Elliott, and Manuela Züger. 2014. Using Psycho-physiological Measures to Assess Task Difficulty in Software Development. In Proceedings of the International Conference on Software Engineering (ICSE). 402–413.
9. Придатко О. В., Придатко В. В., Борзов Ю. О., Дзень В. Є. Інтеграція новаційного методу мобільного навчання в освітні проекти підготовки розробників програмного забезпечення. Вісник ЛДУБЖД, Львів: ЛДУ БЖД, 2018. – №18. – С.70-80. 14. Barraood S. O., Mohd H., Baharom F. A Comparison Study of Software

Testing Activities in Agile Methods. Knowledge Management International Conference (KMICe) Virtual Conference. Malaysia, 2021. pp. 130-137

10. Загальносистемні принципи та етапи створення програм. Життєвий цикл програмного виробу. [Електронний ресурс] – URL: <http://lib.mdpu.org.ua/e-book/vstup/L7.htm>.

11. Моделі життєвого циклу, принципи і методології розробки програмного забезпечення [Електронний ресурс] – URL: <https://evergreens.com.ua/ua/articles/software-development-metodologies.html>.

12. Асєєва А. В., Кулаковська І. В. Аналіз проблем вибору технології для розробки програмного забезпечення. Комп'ютерно-інтегровані технології: освіта, наука, виробництво. Луцьк, 2019. №37. С. 10 – 18, <https://doi.org/10.36910/6775-2524-0560-2019-37-2>.

13. Test Deliverables in Software Testing – Detailed Explanation [Electronic resource] - URL: <https://www.softwaretestingmaterial.com/testdeliverables/>.

14. Авраменко А.С. Тестування програмного забезпечення. Навчальний посібник / А. С. Авраменко, В. С. Авраменко, Г. В. Косенюк. – Черкаси: ЧНУ імені Богдана Хмельницького, 2017. - 284 с

15. Kristien Ooms, Lien Dupont, Lieselot Lapon, and Stanislav Popelka. 2015. Accuracy and precision of fixation locations recorded with the low-cost Eye Tribe tracker in different experimental setups. Journal of eye movement research 8, 1 (2015).

16. Barbareschi M., Barone S., Carbone R. et al. Scrum for safety: an agile methodology for safetycritical software systems. Software Qual J. 2022. №30, pp. 1067–1088 <https://doi.org/10.1007/s11219-022-09593-2>.

17. J. Hofmeister, J. Siegmund, and D. V. Holt. Shorter identifier names take longer to comprehend. In 2017 IEEE 24th International Conference on Software Analysis, Evolution and Reengineering (SANER), pages 217–227, 2017.

18. Motivation and Efficiency of Quality Management Systems Implementation: A Study of Lithuanian Organizations [Електронний ресурс]. - Режим доступу:

https://www.researchgate.net/publication/264857878_Motivation_and_Efficiency_of_Quality_Management_Systems_Implementation_A_Study_of_Lithuanian_Organizations.

19. Ponzanelli, Simone Scalabrino, Gabriele Bavota, Andrea Mocci, Massimiliano Di Penta, Rocco Oliveto, and Michele Lanza. Supporting Software Developers with a Holistic Recommender System. In Proceedings of ICSE 2017 (39th ACM/IEEE International Conference on Software Engineering). to be published, 2017.

20. Sarah Fakhoury, Yuzhan Ma, Venera Arnaoudova, and Olusola Adesope. The effect of poor source code lexicon and readability on developers' cognitive load. In 2018 IEEE/ACM 26th International Conference on Program Comprehension (ICPC), pages 286–28610, 2018.

21. Студентський репозитарій. [Електронний ресурс] URL: <https://studfile.net/preview/6305127/page:2>. <https://studfile.net/preview/6305131/>.

22. Моргун І.А. Метод експертної оцінки якості програмного забезпечення / І.А. Моргун // Матеріали міжнародної науково-практичної конференції аспірантів і студентів «Інженерія програмного забезпечення», №2(6), 2011. С. 33-37.

23. Sacha Lity, Manuel Nieke, Thomas Thüm, Ina Schaefer : Retest test selection for product-line regression testing of variants and versions of variants. Journal of Systems and Software, No.147, January 2019. Pp 46-63.

24. Математичне моделювання процесу розробки спеціалізованих програмних систем безпеко-орієнтованого спрямування /Ю. С. Кордунова, М. Фелтіновські, О. В. Придатко, О. О. Смотр //Вісник ЛДУБЖД Bulletin of Lviv State University of Life Safety <https://journal.ldubgd.edu.ua/index.php/Visnuk> ISSN 2078-4643.

25. Pragati Baheti, «What is a neural network activation function and how does it work? Explore twelve different types of activation functions and learn how to pick the right one». <https://www.v7labs.com/blog/neural-networks-activation-functions>

26. Когут В.С., Метод інтеграції технологій машинного навчання у програмну систему управління бізнесом шляхом точкової автоматизації бізнес-процесів. // Рбірник наукових праць за матеріалами XVI Всеукраїнської науково-практичної конференції «Актуальні проблеми комп'ютерних наук АПКН-2024», 2024, с. 266-268

ДОДАТОК А

(обов'язковий)

ПРОГРАМНИЙ КОД ОСНОВНИХ МОДУЛІВ

A.1 Програмний код мікросервісу, який отримує дані транзакцій із Monobank API, валідує їх та зберігає у базу даних MySQL.

```
DB_CONFIG = {
    "host": "localhost",
    "user": "root",
    "password": "password",
    "database": "payments_db",
}
MONOBANK_API_KEY = ""

class Transaction(BaseModel):
    id: str
    time: int
    description: str
    amount: float
    currency_code: int

def get_db_connection():
    return mysql.connector.connect(**DB_CONFIG)

def fetch_transactions(account_id: str, from_timestamp: int) ->
List[Transaction]:
    url = MONOBANK_API_URL.format(account_id=account_id,
from_timestamp=from_timestamp)
    headers = {"X-Token": MONOBANK_API_KEY}
    response = requests.get(url, headers=headers)

    if response.status_code != 200:
        raise HTTPException(status_code=response.status_code,
detail="Failed to fetch transactions")

    transactions = response.json()
    return [Transaction(**txn) for txn in transactions]

def save_transactions_to_db(transactions: List[Transaction]):
    connection = get_db_connection()
    cursor = connection.cursor()

    query = """
INSERT INTO transactions (id, time, description, amount,
currency_code)
VALUES (%s, %s, %s, %s, %s)
ON DUPLICATE KEY UPDATE
```

```

        description = VALUES(description), amount = VALUES(amount),
currency_code = VALUES(currency_code);
"""

        for txn in transactions:
            cursor.execute(query, (txn.id, txn.time,
txn.description, txn.amount, txn.currency_code))

            connection.commit()
            cursor.close()
            connection.close()

@app.post("/fetch-transactions/{account_id}")
def fetch_and_save_transactions(account_id: str, from_timestamp:
int):
    try:
        transactions = fetch_transactions(account_id,
from_timestamp)
        save_transactions_to_db(transactions)
        return {"message": "Transactions fetched and saved
successfully", "count": len(transactions)}
    except Exception as e:
        raise HTTPException(status_code=500, detail=str(e))

```

A.2 Програмний код мікросервісу для класифікації інвойсів

```
from flask import Flask, request, jsonify
import tensorflow as tf
import numpy as np
import logging
import os

app = Flask(__name__)

logging.basicConfig(level=logging.INFO, format='%(asctime)s -
%(message)s')
logger = logging.getLogger()

MODEL_PATH = "models/invoice_classifier_model"
model = tf.keras.models.load_model(MODEL_PATH)
logger.info(f"Model loaded from {MODEL_PATH}")

# Функція для підготовки фіч
def prepare_features(df):

    scaler = StandardScaler()
    df[['payment_amount', 'invoice_amount']] =
scaler.fit_transform(df[['payment_amount', 'invoice_amount']])

    df['payment_to_invoice_ratio'] = df['payment_amount'] /
df['invoice_amount']

    le = LabelEncoder()
    df['invoice_type'] = le.fit_transform(df['invoice_type'])

    return df[['payment_amount', 'invoice_amount',
'payment_to_invoice_ratio', 'invoice_type']]

def classify_payment(features):
    try:
        features = np.array(features).reshape(1, -1)

        prediction = model.predict(features)
        predicted_class = np.argmax(prediction, axis=1)[0]

        return predicted_class, prediction.tolist()
    except Exception as e:
        logger.error(f"Error in prediction: {e}")
        return None, None

@app.route('/predict', methods=['POST'])
def predict():
    try:
        data = request.get_json()
        if 'features' not in data:
```

```

        return jsonify({"error": "Missing 'features' in request
body"}), 400

    features = prepare_features(pd.DataFrame(data))
    predicted_class, probabilities = classify_payment(features)

    if predicted_class is None:
        return jsonify({"error": "Prediction failed"}), 500

    return jsonify({
        "predicted_class": int(predicted_class),
        "probabilities": probabilities
    })
except Exception as e:
    logger.error(f"Error in /predict: {e}")
    return jsonify({"error": "An unexpected error occurred"}), 500

if __name__ == '__main__':
    HOST = os.getenv('HOST', '0.0.0.0')
    PORT = int(os.getenv('PORT', 5000))
    app.run(host=HOST, port=PORT)

```

A.3 Програмний код мікросервісу для роботи з базою даних

```

import mysql.connector
from mysql.connector import Error
import json

class DatabaseService:
    def __init__(self, host, user, password, database):
        self.host = host
        self.user = user
        self.password = password
        self.database = database
        self.connection = None
        self.cursor = None
        self.connect()

    def connect(self):
        try:
            self.connection = mysql.connector.connect(
                host=self.host,
                user=self.user,
                password=self.password,
                database=self.database
            )
            if self.connection.is_connected():
                self.cursor = self.connection.cursor(dictionary=True)
        except Error as e:
            print(f"Error while connecting to MySQL: {e}")

```

```

def close(self):
    if self.connection.is_connected():
        self.cursor.close()
        self.connection.close()

def insert_invoice(self, invoice_id, user_id, amount, due_date,
status):
    query = """
INSERT INTO invoices (invoice_id, user_id, amount, due_date,
status)
VALUES (%s, %s, %s, %s, %s)
"""
    data = (invoice_id, user_id, amount, due_date, status)
    try:
        self.cursor.execute(query, data)
        self.connection.commit()
    except Error as e:
        print(f"Error inserting invoice: {e}")

def update_invoice_status(self, invoice_id, status):
    query = """
UPDATE invoices
SET status = %s
WHERE invoice_id = %s
"""
    data = (status, invoice_id)
    try:
        self.cursor.execute(query, data)
        self.connection.commit()
    except Error as e:
        print(f"Error updating invoice status: {e}")

def insert_payment(self, payment_id, user_id, amount,
payment_date):
    query = """
INSERT INTO payments (payment_id, user_id, amount,
payment_date)
VALUES (%s, %s, %s, %s)
"""
    data = (payment_id, user_id, amount, payment_date)
    try:
        self.cursor.execute(query, data)
        self.connection.commit()
    except Error as e:
        print(f"Error inserting payment: {e}")

def get_invoices_by_user(self, user_id):
    query = """
SELECT * FROM invoices
WHERE user_id = %s

```

```

"""
try:
    self.cursor.execute(query, (user_id,))
    result = self.cursor.fetchall()
    return result
except Error as e:
    print(f"Error fetching invoices by user: {e}")
    return []

def get_payments_by_user(self, user_id):
    query = """
SELECT * FROM payments
WHERE user_id = %s
"""
    try:
        self.cursor.execute(query, (user_id,))
        result = self.cursor.fetchall()
        return result
    except Error as e:
        print(f"Error fetching payments by user: {e}")
        return []

def get_invoice_by_id(self, invoice_id):
    query = """
SELECT * FROM invoices
WHERE invoice_id = %s
"""
    try:
        self.cursor.execute(query, (invoice_id,))
        result = self.cursor.fetchone()
        return result
    except Error as e:
        print(f"Error fetching invoice by ID: {e}")
        return None

def link_payment_to_invoice(self, payment_id, invoice_id,
payment_amount):
    query = """
INSERT INTO payments_invoices (payment_id, invoice_id,
payment_amount)
VALUES (%s, %s, %s)
"""
    data = (payment_id, invoice_id, payment_amount)
    try:
        self.cursor.execute(query, data)
        self.connection.commit()
    except Error as e:
        print(f"Error linking payment to invoice: {e}")

def get_linked_invoices_for_payment(self, payment_id):
    query = """

```

```

SELECT i.invoice_id, i.amount, pi.payment_amount
FROM invoices i
JOIN payments_invoices pi ON i.invoice_id = pi.invoice_id
WHERE pi.payment_id = %s
"""
try:
    self.cursor.execute(query, (payment_id,))
    result = self.cursor.fetchall()
    return result
except Error as e:
    print(f"Error fetching linked invoices for payment: {e}")
    return []

def save_classification_result(self, user_id, payment_id,
invoice_ids):
    query = """
INSERT INTO classification_results (user_id, payment_id,
invoice_ids)
VALUES (%s, %s, %s)
"""
    invoice_ids_json = json.dumps(invoice_ids)
    data = (user_id, payment_id, invoice_ids_json)
    try:
        self.cursor.execute(query, data)
        self.connection.commit()
    except Error as e:
        print(f"Error saving classification result: {e}")

def get_classification_results(self, payment_id):
    query = """
SELECT * FROM classification_results
WHERE payment_id = %s
"""
    try:
        self.cursor.execute(query, (payment_id,))
        result = self.cursor.fetchone()
        return result
    except Error as e:
        print(f"Error fetching classification results: {e}")
        return None

def update_classification_result(self, payment_id, invoice_ids):
    query = """
UPDATE classification_results
SET invoice_ids = %s
WHERE payment_id = %s
"""
    invoice_ids_json = json.dumps(invoice_ids)
    data = (invoice_ids_json, payment_id)
    try:
        self.cursor.execute(query, data)

```

```

        self.connection.commit()
    except Error as e:
        print(f"Error updating classification result: {e}")

```

A.4 Програмний код мікросервісу тренування моделі

```

from flask import Flask, request, jsonify
import tensorflow as tf
from tensorflow.keras.models import load_model
import numpy as np
import pandas as pd
from sklearn.preprocessing import StandardScaler
import os
import time

app = Flask(__name__)

MODEL_PATH = 'models/current_model.h5'
DATA_PATH = 'data/training_data.csv'
BATCH_SIZE = 32
EPOCHS = 100
LEARNING_RATE = 0.001

def load_data(file_path):
    data = pd.read_csv(file_path)
    return data

def preprocess_data(data):
    features = data.drop(columns=['target'])
    target = data['target']
    scaler = StandardScaler()
    features_scaled = scaler.fit_transform(features)
    return features_scaled, target

def create_model(input_shape):
    model = tf.keras.Sequential([
        tf.keras.layers.Dense(128, activation='relu',
input_shape=input_shape),
        tf.keras.layers.Dense(64, activation='relu'),
        tf.keras.layers.Dense(6, activation='sigmoid')
    ])

model.compile(optimizer=tf.keras.optimizers.Adam(learning_rate=LEARNING_RATE),
              loss='mse', metrics=['mae'])
    return model

def train_model(model, features, target):
    model.fit(features, target, epochs=EPOCHS, batch_size=BATCH_SIZE,
verbose=1)
    return model

```

```

def save_model(model, path):
    model.save(path)

def fine_tune_model(model, features, target):
    model.fit(features, target, epochs=EPOCHS, batch_size=BATCH_SIZE,
    verbose=1)
    return model

@app.route('/train', methods=['POST'])
def train():
    try:
        data = load_data(DATA_PATH)
        features, target = preprocess_data(data)

        if os.path.exists(MODEL_PATH):
            model = load_model(MODEL_PATH)
            model = fine_tune_model(model, features, target)
        else:
            model = create_model((features.shape[1],))
            model = train_model(model, features, target)

        save_model(model, MODEL_PATH)
        return jsonify({"message": "Model trained successfully"}), 200

    except Exception as e:
        return jsonify({"error": str(e)}), 500

@app.route('/retrain', methods=['POST'])
def retrain():
    try:
        new_data = request.get_json()
        new_df = pd.DataFrame(new_data)
        features, target = preprocess_data(new_df)

        if os.path.exists(MODEL_PATH):
            model = load_model(MODEL_PATH)
            model = fine_tune_model(model, features, target)
            save_model(model, MODEL_PATH)
            return jsonify({"message": "Model retrained
successfully"}), 200
        else:
            return jsonify({"error": "Model not found"}), 404

    except Exception as e:
        return jsonify({"error": str(e)}), 500

@app.route('/predict', methods=['POST'])
def predict():
    try:
        input_data = request.get_json()

```

```
input_df = pd.DataFrame(input_data)
input_features = input_df.values

if os.path.exists(MODEL_PATH):
    model = load_model(MODEL_PATH)
    predictions = model.predict(input_features)
    return jsonify({"predictions": predictions.tolist()}),
200
else:
    return jsonify({"error": "Model not found"}), 404

except Exception as e:
    return jsonify({"error": str(e)}), 500

if __name__ == '__main__':
    app.run(debug=True, host='0.0.0.0', port=5000)
```

ДОДАТОК Б
(обов'язковий)

КОПІЇ НАУКОВИХ ПУБЛІКАЦІЙ

МЕТОД ІНТЕГРАЦІЇ ТЕХНОЛОГІЙ МАШИННОГО НАВЧАННЯ У ПРОГРАМНІ СИСТЕМИ УПРАВЛІННЯ БІЗНЕСОМ ШЛЯХОМ ТОЧКОВОЇ АВТОМАТИЗАЦІЇ БІЗНЕС-ПРОЦЕСІВ

Автоматизація бізнес-процесів є важливим компонентом ефективного управління компанією. В умовах сучасної конкуренції підприємства намагаються оптимізувати витрати, підвищувати продуктивність та зменшувати вплив людського фактора на роботу систем. Технології машинного навчання (ML) відіграють ключову роль у цьому процесі, дозволяючи впроваджувати автоматизацію в рутинні та критичні етапи бізнесу. Інтеграція ML у системи управління бізнесом дозволяє автоматизувати рішення, підвищити точність прогнозування та виявити приховані закономірності в даних, що недоступні через традиційні підходи.

Automation of business processes is an important component of effective company management. In the conditions of modern competition, enterprises are trying to optimize costs, increase productivity and reduce the influence of the human factor on the operation of systems. Machine learning (ML) technologies play a key role in this process, enabling automation in routine and critical business steps. Integrating ML into business management systems allows you to automate decisions, increase forecasting accuracy, and reveal hidden patterns in data that are inaccessible through traditional approaches.

Одним із основних викликів, що стоять перед сучасними компаніями, є ефективне управління бізнес-процесами. Наразі існує велика кількість процесів, які потребують або повної автоматизації, або часткової оптимізації, зокрема:

- Рутинні завдання, що виконуються вручну.
- Висока залежність від людського фактора.
- Потреба в швидкому та точному прийнятті рішень.
- Обробка великої кількості даних, що потребує автоматизації для своєчасного аналізу.

Технології машинного навчання можуть допомогти подолати ці проблеми через автоматизацію рутинних завдань, покращення управлінських рішень та ефективне використання ресурсів компанії. Одним із ключових методів є точкова автоматизація процесів, яка зосереджена на автоматизації лише критичних частин бізнес-процесів, що дозволяє зберегти гнучкість і одночасно підвищити ефективність.

Розглянемо приклад автоматизації прогнозування попиту на продукцію. Машинне навчання може використовуватися для аналізу історичних даних продажів, сезонних коливань і ринкових факторів для точнішого передбачення попиту. У таблиці нижче показано порі

вняння традиційних методів прогнозування з методами машинного навчання.

Таблиця 1 – Порівняння традиційних методів і методів машинного навчання

Метод прогнозування	Середня похибка прогнозу, %
Традиційні методи	14
Методи машинного навчання	6

Як показано в таблиці 1, використання технологій машинного навчання дозволяє суттєво знизити похибку прогнозу, що безпосередньо впливає на ефективність управління запасами та скорочення витрат.

Перелік посилань

- 6 Smith, J. **Business Process Automation with Machine Learning: A Practical Guide**. New York: Springer, 2020.
- 7 Lee, Y. **Machine Learning in Business Process Management: An Integrated Approach** [Online Resource]. – Access Mode: <https://www.springer.com/business-process-management>.
- 8 Johnson, R. **AI and Machine Learning for Business Automation** [Electronic Resource]. – Access Mode: <https://www.ai-business.com/ml-automation>.
- 9 Thompson, A. **Optimizing Business Processes Using Machine Learning**. – London: Wiley, 2021.

ДОДАТОК В

(обов'язковий)

ПРЕЗЕНТАЦІЙНІ МАТЕРІАЛИ

Хмельницький Національний Університет
Факультет інформаційних технологій

Кафедра інженерії програмного забезпечення

Метод інтеграції технологій машинного навчання у програмну систему управління бізнесом шляхом точкової автоматизації бізнес-процесів

Виконав
студент II курсу, групи ІПЗм-23-1
Когут Василь Сергійович

Керівник
Бедратюк Леонід Петрович
доктор фіз.-мат. наук, професор

2024

АКТУАЛЬНІСТЬ ТЕМИ

Актуальність роботи полягає у створенні методів точкової автоматизації на основі технологій машинного навчання, що дозволяють адаптуватися до ринкових змін і забезпечувати ефективне управління ресурсами. Сучасні системи автоматизації управління бізнесом часто побудовані на основі статичних алгоритмів, що ускладнює їхнє застосування в умовах динамічних ринкових змін. **Використання машинного навчання** дозволяє створювати більш гнучкі системи, здатні адаптуватися до нових вимог та змін у реальному часі. Таким чином, необхідно розробити метод інтеграції ML, що дозволить використовувати точкову автоматизацію, зосереджуючись лише на ключових процесах, які потребують оптимізації.



МЕТА І ЗАВДАННЯ ДОСЛІДЖЕННЯ



Метою роботи є розроблення методу інтеграції машинного навчання у програмні системи управління бізнесом для забезпечення адаптивної точкової автоматизації бізнес-процесів і оптимізації ключових управлінських етапів

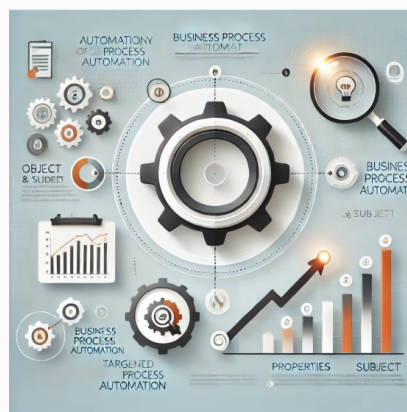
Завдання даного дослідження проаналізувати наявні методи, ідентифікувати критичні точки де провадження ML буде максимально ефективним, розробка алгоритмів для конкретних бізнес - процесів, інтеграція моделі у систему управління бізнесом і оцінка ефективності розробленого методу

3

ОБ'ЄКТ І ПРЕДМЕТ ДОСЛІДЖЕННЯ

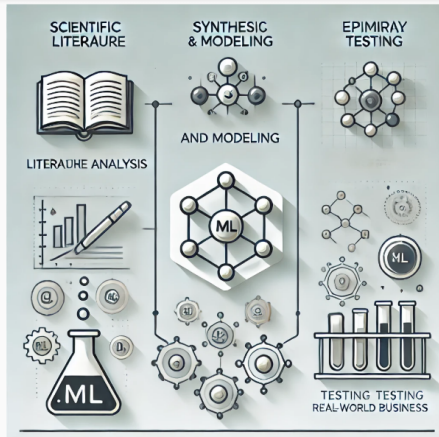
Об'єктом дослідження є процеси автоматизації бізнес-процесів у програмних системах управління

Предметом дослідження є методи інтеграції технологій машинного навчання для точкової автоматизації бізнес-процесів, спрямовані на підвищення ефективності управлінських рішень



4

МЕТОДИ ДОСЛІДЖЕННЯ



Для досягнення мети використані наступні методи: аналіз наукової літератури для вивчення поточного стану дослідження в галузі ML і бізнес автоматизації, синтез і моделювання для розробки алгоритмів ML, емпіричне тестування яке дозволило перевірити ефективність розробленого методу у реальних бізнес - сценаріях

5

НАУКОВА НОВИЗНА ОТРИМАНИХ РЕЗУЛЬТАТІВ

Наукова новизна отриманих результатів полягає в тому що вперше розроблено метод інтеграції машинного навчання у програмні системи управління бізнесом, який дозволяє здійснювати точкову автоматизацію бізнес-процесів для підвищення адаптивності до змін ринкових умов. Також удосконалено підходи до адаптації моделей машинного навчання у бізнес-середовищах, що забезпечують оптимізацію процесів, які потребують інтенсивної обробки даних



6

АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ



Розділ підсумовує існуючі рішення у сфері автоматизації бізнес-процесів з використанням машинного навчання (ML). Огляд показав, що сьогодні широко застосовуються CRM-системи (наприклад, Salesforce Einstein, Zoho CRM) для управління взаємодією з клієнтами, а також спеціалізовані платформи для автоматизації бізнес-процесів, такі як UiPath та Automation Anywhere, що використовують AI для обробки великих обсягів даних та забезпечення ефективного керування процесами.

Існуючі рішення, незважаючи на широкий функціонал і можливості, мають свої обмеження, зокрема у складності інтеграції, масштабованості та необхідності в людському контролі у нестандартних ситуаціях. Це вказує на необхідність подальших досліджень для розробки більш адаптивних і всеосяжних систем,

Після дослідження джерел можна виділити такі недоліки та переваги традиційних систем та систем з ML

Категорія	Традиційні EDM системи	Системи з ML
Якість даних	Ручне введення даних призводить до <u>непослідовностей</u> та помилок.	Фрагментарна автоматизація може не вирішити всіх проблем якості даних.
Масштабованість	Обмежена масштабованість зі збільшенням обсягів даних.	Масштабованість залежить від інфраструктури та складності моделі.
Робота в режимі реального часу	Пакетна обробка обмежує прийняття рішень в режимі реального часу.	Можливості в режимі реального часу можуть варіюватися, вимагаючи додаткової інтеграції.
Інтеграція	Розрізнені сховища даних ускладнюють доступ до міжфункціональних інсайтів.	Проблеми інтеграції з різними джерелами даних.
Складність	Спирається на алгоритми, засновані на правилах; бракує адаптивного навчання.	Потребує експертизи в галузі науки про дані та розгортання моделей ML.
Вартість	Низькі початкові інвестиції, але вищі операційні витрати.	Вищі початкові інвестиції через інфраструктуру та навчання.

МЕТОД ІНТЕГРАЦІЇ МАШИННОГО НАВЧАННЯ ДЛЯ АВТОМАТИЗАЦІЇ БІЗНЕС-ПРОЦЕСІВ

Робота проводилась на базі реальної бізнес системи, що використовується на спортивних секціях. Тому ми мали набір реальних даних, короткий опис якого є в таблиці. Ця система має проблему, яка полягає в тому, що вони часто отримують платежі, де не завжди чітко вказано, які саме рахунки вони покривають. У результаті розподіл платежів між рахунками здійснюється вручну, що займає час і є схильним до помилок, тому потребує автоматизації.

У розділі 2 був огляд методів машинного навчання для автоматизованого розподілу платежів та методів інтеграції у систему

Характеристика	Значення
Кількість платежів	5,000
Кількість користувачів	155
Період даних	6 місяців
Середня кількість платежів на клієнта	32
Мета даних	Класифікація інвойсів для кожного платежу
Попередня обробка	Нормалізація, кодування категорій, обробка пропусків
Цільова змінна	Інвойси, покриті кожним платежем

9

МЕТОД ІНТЕГРАЦІЇ МАШИННОГО НАВЧАННЯ ДЛЯ АВТОМАТИЗАЦІЇ БІЗНЕС-ПРОЦЕСІВ



Для вирішення цієї задачі було прийнято рішення тренувати нейронну мережу прямого поширення. Вибрані шари з активаційними функціями ReLU та Sigmoid.

Метрики для оцінки нейронної мережі: Точність (accuracy): Оцінює, яку частину прогнозів нейронна мережа зробила правильно. Precision та Recall: Для вимірювання того, скільки інвойсів було правильно передбачено як покриті платіжною сумою, і скільки помилок або пропусків сталося. F1-Score: Сумарна метрика, яка є корисною для оцінки балансу між точністю та повнотою на основі класів, де важливо уникати як пропусків, так і помилок.

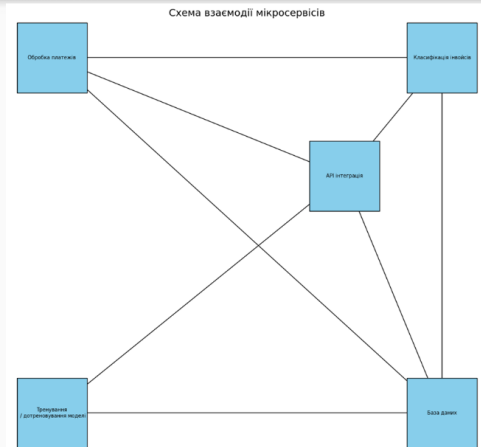
10

АРХІТЕКТУРА ПРОГРАМНОЇ РЕАЛІЗАЦІЇ

У розділі 3 формувались вимоги і вибір архітектури програмної реалізації та вимоги до нейронної мережі.

Так як одна мета цієї роботи це розробка методу інтеграції то зверталась велика увага на вибір архітектури, яка не мала б мати тих проблем що є у наявних рішеннях.

По цій причині було вибрано мікросервісну архітектуру, що дозволяє просто змінювати різні частини системи без переписування інших. Також одна з вимог була можливість дотреноувати/перетреноувати нейронну мережу на нових даних, тому під це був зроблений ще один окремий сервіс.



11

АРХІТЕКТУРА ПРОГРАМНОЇ РЕАЛІЗАЦІЇ

Основні мікросервіси:

1. Обробка платіжних транзакцій.
2. Класифікація інвойсів.
3. Робота з базою даних.
4. Навчання моделі та API для взаємодії з іншими частинами системи

12

ПРОГРАМНА РЕАЛІЗАЦІЯ

У розділі 4 представлена сама програмна реалізація та результати тренування та тестування моделі і системи.

Середня точність моделі під час тренування була 94%. Під час тестування 92%.

Однією з основних переваг цієї інтеграції є те, що наше рішення базується на мікросервісній архітектурі, що дозволяє гнучко масштабувати систему відповідно до зростаючих обсягів даних і змінюваних вимог. Це було підтверджено тестами які показали що ми можемо обробляти до 10 тисяч запитів за хвилину, що у разі швидше чим ручна обробка. Мікросервіси можуть працювати незалежно один від одного, що забезпечує більшу стабільність і надійність системи в цілому.

13

ПУБЛІКАЦІЇ

Конференція АПКН-2024

МЕТОД ІНТЕГРАЦІЇ ТЕХНОЛОГІЙ МАШИННОГО
НАВЧАННЯ У ПРОГРАМНІ СИСТЕМИ
УПРАВЛІННЯ БІЗНЕСОМ ШЛЯХОМ ТОЧКОВОЇ
АВТОМАТИЗАЦІЇ БІЗНЕС-ПРОЦЕСІВ

У збірнику с. 266-268



14

ВИСНОВКИ

У процесі дослідження було здійснено аналіз існуючих методів, що дозволило виявити критичні точки, де впровадження машинного навчання забезпечить максимальну ефективність. Було розроблено алгоритми, зокрема використовуючи нейронні мережі, для точного прогнозування покриття платежів, а також здійснено інтеграцію розробленої моделі в існуючу систему управління бізнесом, що дозволило оптимізувати фінансові потоки.

Закриті завдання:

- Аналіз наявних методів і ідентифікація критичних точок для впровадження ML.
- Розробка алгоритмів для конкретних бізнес-процесів.
- Інтеграція моделі в систему управління бізнесом.
- Оцінка ефективності розробленого методу.

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Здобувач Когут Василь Сергійович

Тема Метод інтеграції технологій машинного навчання у програмні системи бізнесом шляхом точкової автоматизації бізнес-процесів

Спеціальність 121 «Інженерія програмного забезпечення»

Обсяг кваліфікаційної роботи:

Кількість сторінок кваліфікаційної роботи: 114 с.

1. Короткий зміст роботи та прийнятих рішень: У кваліфікаційній роботі здійснено системний аналіз предметної області у сфері автоматизації бізнес-процесів із використанням технологій машинного навчання, а також виявлено проблемні аспекти та невирішені питання в інтеграції ML у бізнес-системи. На основі проведеного аналізу запропоновано, розроблено та програмно реалізовано метод інтеграції машинного навчання у програмні системи управління бізнесом з акцентом на точкову автоматизацію бізнес-процесів. Завдяки використанню моделей нейронних мереж реалізовано систему, що дозволяє розподіляти платежі та оптимізувати фінансові потоки. Отримані результати підтверджують доцільність та ефективність розробленого підходу, сприяючи підвищенню гнучкості, точності та швидкості автоматизованих бізнес-систем.

2. Висновок про відповідність роботи поставленому завданню Кваліфікаційна робота освітнього ступеня «магістр» у повній мірі відповідає поставленому завданню як у теоретичній, так і в практичній її частині.

3. Характеристика виконання кожного розділу роботи, ступінь використання останніх досягнень науки і техніки і передових методів роботи: У вступі обґрунтовано актуальність теми дослідження, сформульовано мету та завдання роботи, визначено об'єкт, предмет, наукову новизну та практичне значення отриманих результатів. У першому розділі виконано аналіз предметної області у сфері інтеграції технологій машинного навчання у бізнес-системи, розглянуто сучасні підходи до автоматизації бізнес-процесів, зокрема із застосуванням ML. У другому розділі розглянуто розробку методу інтеграції машинного навчання для точкової автоматизації бізнес-процесів. На основі аналізу визначено критичні точки і запропоновано алгоритми, адаптовані для вирішення специфічних бізнес-задач. У третьому розділі обґрунтовано архітектурні рішення системи управління бізнесом, що дозволяють інтегрувати моделі машинного навчання. Особливу увагу приділено використанню мікросервісної архітектури для забезпечення масштабованості та сумісності компонентів. У четвертому розділі описано реалізацію програмної системи, що базується на розроблених методах і алгоритмах. Проведено тестування запропонованого підходу у реальних бізнес-сценаріях, оцінено його ефективність у процесах оптимізації фінансових потоків. Результати підтвердили доцільність і ефективність розробленого методу. Робота виконана із використанням сучасних досягнень науки та техніки в галузі машинного навчання, мікросервісної архітектури та аналізу бізнес-процесів.

4. Позитивні сторони роботи Кваліфікаційна робота містить низку інноваційних рішень, спрямованих на інтеграцію машинного навчання у програмні системи

управління бізнесом. Запропоновані алгоритми автоматизації фінансових операцій забезпечують високу точність і швидкість обробки даних, мінімізують людський фактор. Ефективність запропонованого методу підтверджена результатами апробації, що продемонстрували підвищення точності прийняття управлінських рішень.

5. Негативні сторони роботи У роботі використано всього один підхід до інтеграції машинного навчання у бізнес-систему, що не дозволяє врахувати різні аспекти автоматизації. Можливо, доцільніше було б зосередитися на вивченні двох ключових алгоритмів із глибшим аналізом їхніх переваг та недоліків. Також можна зазначити, що обмеження даних, доступних для навчання моделей, може вплинути на універсальність результатів. Більше уваги слід приділити тестуванню запропонованих рішень у ширшому діапазоні бізнес-сценаріїв, що забезпечило б більш ґрунтовну оцінку їхньої ефективності.

6. Оцінка графічного оформлення та пояснювальної записки роботи: Графічне оформлення виконане відповідно до теми кваліфікаційної роботи з дотриманням вимог стандартів. Пояснювальна записка відповідає вимогам стандартів до її оформлення.

7. Відгук про роботу в цілому В цілому кваліфікаційна робота заслуговує на високу позитивну оцінку. Робота вирізняється чіткою структурою, логічною послідовністю та високим рівнем наукової аргументації. Усі розділи гармонійно пов'язані між собою, забезпечуючи системний підхід до розкриття теми. Графічний матеріал є якісним і наочним, він ефективно доповнює текстову частину, демонструючи доцільність і результативність запропонованих рішень. Загалом робота свідчить про високий рівень професійної підготовки автора та здатність до вирішення складних наукових і практичних завдань.

8. Інші зауваження

9. Оцінка кваліфікаційної роботи Оцінка кваліфікаційної роботи Розглянувши позитивні та негативні сторони представленої кваліфікаційної роботи, можна зробити висновок, що вона заслуговує оцінки «відмінно».

РЕЦЕНЗЕНТ (прізвище, ім'я, по-батькові, посада, місце роботи)

*Мартишок Валерій Володимирович,
д.т.н., професор, зав. кафедр. АІТІ та Р*

«26» 11 2024 р.

(підпис)

Протокол аналізу звіту подібності науковим керівником

Заявляю, що я ознайомився (-лась) з Повним звітом подібності, який був згенерований Системою виявлення і запобігання плагіату щодо роботи:

Автор: Василь Когут

Співавтор:

Назва: МКР Метод інтеграції технологій машинного навчання у програмну систему управління бізнесом шляхом точкової автоматизації бізнес-процесів

Науковий керівник: д-р фіз.-мат. наук, професор Леонід БЕДРАТЮК

Підрозділ: Кафедра інженерії програмного забезпечення

Коефіцієнт подібності 1: 3.1%

Коефіцієнт подібності 2: 1.9%

Мікропробіли: 22

Заміна букв: 1

Інтервали: 0

Білі знаки: 0

Дата створення звіту: 2024-11-28 01:42:02.0

Після аналізу Звіту подібності констатую наступне:

- Запозичення, виявлені в роботі є законними і не є плагіатом. Рівень подібності не перевищує допустимої межі. Таким чином робота незалежна і приймається.
- Запозичення не є плагіатом, але перевищено граничне значення рівня подібностей. Таким чином робота повертається на доопрацювання.
- Виявлено запозичення і плагіат або навмисні текстові спотворення (маніпуляції), як передбачувані спроби укриття плагіату, які роблять роботу невідповідною вимогам законодавства (Ст. 32. ЗУ Про вищу освіту, пункт 3.1, Ст. 42. ЗУ Про освіту) та вимог НАЗЯВО (Критерій 5), а також кодексу етики і процедурам. Таким чином робота не приймається.

Обґрунтування:

28.11.2024
Дата


експерт

Anti-Plagiarism v-15.257

Максимальне співпадіння з одним документом 2.0%

Словники перевірки: en_US, ru_RU, ua_UA. Помилки в документах: 8%

ID: 149764 Назва: КВАЛІФІКАЦІЙНА РОБОТА Метод інтеграції технологій машинного навчання у програмну систему управління бізнесом шляхом точкової автоматизації бізнес-процесів Додано в БД: 2024-11-25 Автора: КОГУТ Василь Керівники: д-р фіз.-мат. наук, професор Леонід БЕДРАТЮК Консультанти: Опоненти:	Документ		Сумарний збіг по Базі Даних	
	Символи	Лексеми	Символи	Лексеми
	167094	1237	6988 (4%)	84 (7%)

Джерело плагіату

ID	Опис	Наявність плагіату в документі	
		Символи	Лексеми

Завідувачу кафедри
інженерії програмного забезпечення
проф. Леоніду БЕДРАТЮКУ
студента групи ІПЗм-23-1

Василя Колупа
Ім'я, ПРІЗВИЩЕ

ЗАЯВА

Прошу закріпити за мною тему кваліфікаційної роботи освітнього ступеня «магістр» за спеціальністю 121 «Інженерія програмного забезпечення»:

Метод інтеграції технологій машинного
навчання у програмну систему управління бізнесом
з метою покращення автоматизації бізнес-процесів

(керівник кваліфікаційної роботи – Леонід Бедратюк)
Ім'я, ПРІЗВИЩЕ

07.09.24
Дата

В.С. Колупа
Підпис здобувача

Завідувачу кафедри інженерії програмного
забезпечення проф. Леоніду БЕДРАТЮКУ
здобувача вищої освіти
Василя КОГУТА
факультет ІТ, 2 курс, група ІПЗм-23-1

ЗАЯВА

З правилами чинного Положення про систему забезпечення академічної доброчесності в Хмельницькому національному університеті, згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту і застосування заходів дисциплінарної та академічної відповідальності, ознайомлений. Про використання програмно-технічних засобів для перевірки кваліфікаційних робіт здобувачів вищої освіти на плагіат оповіщений та надаю свою згоду на обробку й збереження університетом моєї роботи в інституційному репозитарії університету.

Також надаю університету право на передачу моєї роботи для обробки та збереження в базах даних програмно-технічних засобів (StrikePlagiarism та Anti-Plagiarism) та використання роботи для виявлення плагіату в інших роботах, які перевіряються програмно-технічними засобами та користувачами, що мають доступ до цих програмно-технічних засобів, виключно в обмежених цілях для виявлення плагіату в текстах робіт.

Робота для перевірки університетом надається в друкованому та електронному варіанті. Електронна версія моєї роботи збігається (ідентична) з друкованою.

21.11.24
дата

В.С.Когута
підпис

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

ДЕКЛАРАЦІЯ УЧАСНИКА ОСВІТНЬОГО ПРОЦЕСУ
щодо дотримання академічної доброчесності

Цією декларацією я, Когут Василь Сергійович

Прізвище, ім'я, по батькові

здобувач вищої освіти (шифр та назва спеці-ті, курс, академічна група/у науково-педагогічний працівник (назва кафедри)

назва факультету

підтверджую, що ознайомився (- лась) з Положенням про систему забезпечення академічної доброчесності в Хмельницькому національному університеті та Кодексом академічної доброчесності і **зобов'язуюсь** дотримуватися їх вимог під час освітнього процесу, проведення наукової діяльності, виконання організаційно-адміністративних функцій тощо.

Усвідомлюю, що у разі порушення мною принципів академічної доброчесності нестиму відповідальність перед академічною спільнотою ХНУ згідно з нормами, визначеними Положенням систему забезпечення академічної доброчесності в Хмельницькому національному університеті, законодавства України.

«02» вересня 2023 р.

В.С. Когут

Підпис